Computers and Mathematics with Applications 127 (2022) 48–64

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

The INTERNODES method for applications in contact mechanics and dedicated preconditioning techniques

Yannis Voet [a,*], Guillaume Anciaux [a], Simone Deparis [b], Paola Gervasio [c]

[a] LSMS, Institute of Civil Engineering, École polytechnique fédérale de Lausanne, Station 18, CH-1015, Lausanne, Switzerland
[b] SCI-SB-SD, Institute of Mathematics, École polytechnique fédérale de Lausanne, Station 8, CH-1015 Lausanne, Switzerland
[c] DICATAM, Università degli Studi di Brescia, via Branze 38, 25123 Brescia, Italy

ARTICLE INFO

Keywords:
Preconditioning
Numerical linear algebra
Finite element method
INTERNODES method
Computational contact mechanics

ABSTRACT

The mortar finite element method is a well-established method for the numerical solution of partial differential equations on domains displaying non-conforming interfaces. The method is known for its application in computational contact mechanics. However, its implementation remains challenging as it relies on geometrical projections and unconventional quadrature rules. The INTERNODES (INTERpolation for NOn-conforming DEcompositionS) method, instead, could overcome the implementation difficulties thanks to flexible interpolation techniques. Moreover, it was shown to be at least as accurate as the mortar method making it a very promising alternative for solving problems in contact mechanics. Unfortunately, in such situations the method requires solving a sequence of ill-conditioned linear systems. In this paper, preconditioning techniques are designed and implemented for the efficient solution of those linear systems.

1. Introduction

Contact mechanics is about the interaction of bodies as they move close to each other. In this paper, the linear elasticity theory will be used to find the deformed configuration of bodies coming into contact. Contrary to single-body elasticity problems, contact problems are intrinsically nonlinear even for the simplest linear constitutive model. The nonlinearity is tied to the unknown contact interface and the resulting inequality constraints. Contact problems are extremely challenging mathematically and we must emphasize that the existence and uniqueness of a solution has only been proved in special cases. Yet, a wide range of computational methods has been developed to meet the industry needs. Unsurprisingly, the inequality constraints tightly bound computational contact mechanics with optimization. In addition, finite element discretizations are typically used to approximate the infinite-dimensional variational problem. Yet again, additional challenges arise in the computations if compared with a single-body elasticity problem. Indeed, different bodies lead to different domains and the solution to the underlying partial differential equation must be coupled across them. This task is hindered by a priori independent discretizations of the bodies, which lead to nonconforming meshes at the interface. The underlying type of nonconformity is inherently geometric, meaning there may exist small gaps or overlaps between the two discrete subdomains. The issue has been traditionally addressed using the mortar finite element method [1–3]. It is based on projection techniques for transferring information between the interfaces. However, this method is known to be difficult to implement and requires ad hoc strategies, for instance to ensure sufficiently accurate numerical quadrature rules and for the special treatment of cross-points when more than two subdomains meet. The INTERNODES (INTERpolation for NOn-conforming DEcompositionS) method [4] appears as a very promising alternative for solving problems in contact mechanics. It is a flexible interpolation based method which overcomes many of the implementation issues of the mortar method and was shown to be at least as accurate [5]. In a recent paper [6], the authors have shown that INTERNODES, like the Mortar method, allows to approximate the conservation of specific quantities, namely that both total force and total work generated by the numerical solution at the interface of the decomposition vanish in an optimal way when the mesh size tends to zero. Preliminary work in computational contact mechanics showed that the INTERNODES method could be successfully applied but also revealed several challenges in efficiently solving the sequence of linear systems arising from the method [7].

In this work, we investigate preconditioning techniques for solving these linear systems. We propose an efficient preconditioner which

* Corresponding author.
E-mail addresses: yannis.voet@epfl.ch (Y. Voet), guillaume.anciaux@epfl.ch (G. Anciaux), simone.deparis@epfl.ch (S. Deparis), paola.gervasio@unibs.it (P. Gervasio).

https://doi.org/10.1016/j.camwa.2022.09.019
Received 2 December 2021; Received in revised form 22 August 2022; Accepted 24 September 2022
Available online 6 October 2022

achieves a very fast convergence independently of the mesh size. The performance of the method relies on a single matrix factorization and targets medium size applications. While the method performs best for exact factorizations, it still performs well for inexact ones. Although the preconditioner can be used for much larger applications, different methods must be sought for solving linear systems with the preconditioning matrix. Large 3D applications and related solution techniques are left as future work. The remainder of the paper is structured as follows. Interpolation being one of the key features of the INTERNODES method, Section 2 provides the necessary background. The emphasis is set on a specific type of interpolation based on radial basis functions. The section extends the work of [8] and discusses implementation aspects. Section 3 covers the mathematical modeling of contact problems in its most basic form. A linear elastic constitutive model is assumed for simplicity. From the strong form of the differential problem, the weak form is derived followed by a finite element discretization. This section covers the application of the INTERNODES method to contact problems. Section 4 gathers some of the properties of the INTERNODES matrix which is an essential step towards preconditioning. Section 5 presents preconditioning techniques tailored to the INTERNODES matrix. Section 6 presents some numerical experiments illustrating the effectiveness of the developed preconditioner. Finally, Section 7 summarizes our findings and discusses issues and prospects.

2. Radial basis function interpolation

Interpolation is at the heart of the INTERNODES method and distinguishes it from mortar finite element methods which are based on projection techniques. In the INTERNODES method, interpolation can be either based on Lagrange or radial basis function (RBF) interpolants. However, RBF interpolants are preferred for problems with geometric non-conformities [4] (geometric non-conformities occur when the two meshes are not watertight at the common interface and small holes and overlaps may be present), and since it is the situation encountered in contact mechanics problems, this section gives the necessary background on RBF. A part of the discussion extends the work in [8] where several modifications to the classical RBF interpolation were introduced. RBF interpolation is especially popular for the interpolation of scattered data and has found applications in particular in neural networks [9] and the numerical solution of partial differential equations [10,11]. For more details on radial basis functions, we refer the reader to the survey paper [12], where both globally and locally supported radial basis functions are presented with an overview of their theoretical properties including convergence of the interpolant to the true function in a neighborhood of the interpolation point.

In the first part of this section, a short introduction to RBF interpolation is given and the modifications introduced in [8] are discussed. Necessary conditions for the modifications to be well defined were already established by the same authors. Sufficient conditions are now derived and implementation aspects are discussed to ensure that the interpolation matrices satisfy such sufficient conditions.

2.1. An introduction to radial basis function interpolation

Let $\Xi = \{\boldsymbol{\xi}_m\}_{m=1}^M$ be a set of interpolation points where some function evaluations are known. We define the global interpolant $\Pi_f(\mathbf{x})$ of a function $f : \mathbb{R}^d \to \mathbb{R}$ as

$$\Pi_f(\mathbf{x}) = \sum_{m=1}^{M} \gamma_m^f \phi(\|\mathbf{x} - \boldsymbol{\xi}_m\|, r),$$

where $\gamma_m^f \in \mathbb{R}$ for $m = 1, \dots, M$ are coefficients and ϕ is a radial basis function which depends on the euclidean distance $\|\mathbf{x} - \boldsymbol{\xi}_m\|$ from the interpolation point $\boldsymbol{\xi}_m$ and is parameterized by a radius r. Typical choices for radial basis functions are listed in Table 1. In the table, the definition of ϕ is given as a function of the normalized distance $\delta = \frac{\|\mathbf{x}-\boldsymbol{\xi}_m\|}{r}$, while $(1-\delta)_+ = \max\{0, 1-\delta\}$.

Table 1
Common radial basis functions.

Name	ϕ
Thin-plate splines	$\delta^2 \ln \delta$
Inverse multiquadratic	$\frac{1}{r\sqrt{\delta^2+1}}$
Gaussian splines	$e^{-\delta^2}$
Wendland C^2	$(1-\delta)_+^4(1+4\delta)$

Thin-plate splines, inverse multiquadratic and Gaussian splines are all globally supported whereas the Wendland C^2 radial basis functions are locally supported. A compact support is very interesting as it leads to sparse interpolation matrices.

Let $\mathbf{f}_\xi \in \mathbb{R}^M$ be the vector containing function evaluations at the interpolation points $\{f(\boldsymbol{\xi}_m)\}_{m=1}^M$ and $\boldsymbol{\gamma}^f \in \mathbb{R}^M$ be the interpolation coefficients. The interpolation conditions $\Pi_f(\boldsymbol{\xi}_m) = f(\boldsymbol{\xi}_m)$ for $m = 1, \dots, M$ lead to the linear system

$$\Phi_{MM}\boldsymbol{\gamma}^f = \mathbf{f}_\xi,$$

where $\Phi_{MM} \in \mathbb{R}^{M\times M}$ is such that $(\Phi_{MM})_{ij} = \phi(\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|, r)$. The radius r being common to all interpolation points, the resulting coefficient matrix Φ_{MM} is symmetric. In addition, for many radial basis functions of practical interest, Φ_{MM} is also positive definite and thus the solution to the linear system is unique. This is the case when using for instance Gaussian or inverse multiquadratic radial basis functions. This property is also satisfied for the Wendland C^2 basis functions [13]. However, this is not the case for thin-plate splines, where a low degree polynomial term must be added to ensure that the interpolation coefficients can be uniquely computed [12]. Moreover, radial basis functions that take only nonnegative values generate nonnegative matrices (matrices with only positive or zero entries).

Let $\Lambda = \{\boldsymbol{\zeta}_n\}_{n=1}^N$ be a set of points where the interpolant is to be evaluated. Let $\mathbf{f}_\zeta \in \mathbb{R}^N$ be the vector containing the evaluations $\{\Pi_f(\boldsymbol{\zeta}_n)\}_{n=1}^N$. Then

$$\mathbf{f}_\zeta = \Phi_{NM}\boldsymbol{\gamma}^f = \Phi_{NM}\Phi_{MM}^{-1}\mathbf{f}_\xi$$

and $\Phi_{NM}\Phi_{MM}^{-1}$ defines an interpolation matrix. In [8] two modifications were introduced:

1. A localized radius was chosen for the radial basis functions. Thus,

   $$(\Phi_{MM})_{ij} = \phi(\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|, r_j) \quad i, j = 1, \dots, M$$
   $$(\Phi_{NM})_{ij} = \phi(\|\boldsymbol{\zeta}_i - \boldsymbol{\xi}_j\|, r_j) \quad i = 1, \dots, N, \quad j = 1, \dots, M.$$

   Using a localized radius allows to take advantage of a nonuniform distribution of interpolation points. In regions where the density of points is high, the radius can be reduced without affecting much the accuracy and allows to spare storage for the interpolation matrices. However, the localized radius unfortunately destroys the symmetry of the matrix Φ_{MM} and the arguments used to prove the positive definiteness of the matrix do not readily extend.
2. A rescaling of the radial basis functions was introduced which enables the exact interpolation of constant functions. Let $\mathbf{1}_\xi \in \mathbb{R}^M$ be a vector containing only ones. We define

   $$\hat{\mathbf{f}}_\zeta = \Phi_{NM}\Phi_{MM}^{-1}\mathbf{f}_\xi \text{ and } \mathbf{g}_\zeta = \Phi_{NM}\Phi_{MM}^{-1}\mathbf{1}_\xi$$

   and set $f_{\zeta_i} = \frac{\hat{f}_{\zeta_i}}{g_{\zeta_i}} \quad i = 1, 2, \dots, N$. This rescaling is in fact equivalent to defining a new interpolant $\bar{\Pi}_f(\mathbf{x}) = \frac{\Pi_f(\mathbf{x})}{\Pi_1(\mathbf{x})}$. Thus, if Π was initially a polynomial function, then $\bar{\Pi}$ is a rational function. On the algebraic side, the rescaling is equivalent to defining the interpolation matrix

   $$R_{NM} = D_{NN}^{-1}\Phi_{NM}\Phi_{MM}^{-1}$$

where $D_{NN} = \text{diag}(\mathbf{g}_\zeta)$ is a diagonal matrix formed by the components of the vector $\mathbf{g}_\zeta$. The matrix R_{NM} will be used extensively throughout the next sections. The numerical results presented in [8] showed that the rescaling greatly improved the accuracy of the interpolation. Yet, the rescaling is only possible if none of the components of the vector $\mathbf{g}_\zeta$ is zero.

Therefore, to assert well-posedness of the interpolation, we must find conditions which guarantee that:

1. The matrix Φ_{MM} is invertible.
2. The vector $\mathbf{g}_\zeta$ does not contain a zero entry.

Clearly, a necessary condition for the second requirement is that Φ_{NM} does not contain a row of zeros. It is equivalent to ensuring that each evaluation point $\boldsymbol{\zeta}_n$ for $n = 1, 2, \dots, N$ lies in the support of at least one basis function. In particular, the second requirement is met if all components of $\mathbf{g}_\zeta$ are strictly positive. We will now find sufficient conditions which not only guarantee the invertibility of Φ_{MM} but also the aforementioned property.

2.2. Sufficient conditions for the rescaled-localized radial basis functions to be well-defined

We will first find a sufficient condition for the matrix Φ_{MM} to be invertible. Then, we will show that the second requirement is also satisfied without further assumptions for some important classes of radial basis functions, including Wendland C^2 and Gaussian splines. One of the key concepts of this section is the one of (strict) diagonal dominance. A matrix $A \in \mathbb{C}^{n\times n}$ is said to be strictly diagonally dominant by rows if

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}| \quad i = 1, 2, \dots, n.$$

An analogous definition exists for strictly diagonally dominant matrices by columns. The proof of the next theorem can be found for example in [14].

Theorem 2.1 *(Levy–Desplanques theorem). Let $A \in \mathbb{C}^{n\times n}$ be a strictly diagonally dominant matrix by rows or columns. Then, A is invertible.*

Consequently, ensuring that Φ_{MM} is strictly diagonally dominant by rows is sufficient for its invertibility. Now, we are particularly interested in knowing something about $\boldsymbol{\gamma} = \Phi_{MM}^{-1}\mathbf{1}_\xi$.

Theorem 2.2. *Let $\boldsymbol{\gamma}$ be the solution of $A\boldsymbol{\gamma} = \mathbf{1}$ where $A \in \mathbb{R}^{n\times n}$ is a nonnegative strictly diagonally dominant matrix by rows with unit diagonal entries ($a_{ii} = 1$ for $i = 1, \dots, n$) and $\mathbf{1}$ is the vector of all ones. Then,*

$$0 < \gamma_i \leq 1 \quad i = 1, 2, \dots, n.$$

Proof. We begin by proving a few useful implications. Considering the ith equation of $A\boldsymbol{\gamma} = \mathbf{1}$, we have

$$\gamma_i + \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\gamma_j = 1 \quad i = 1, \dots, n. \tag{1}$$

Let us investigate different cases:

- Assume that $\gamma_i > 1$, then, from equation (1), $\underbrace{\gamma_i}_{>1} + \underbrace{\sum_{j=1,\ j\neq i}^{n} a_{ij}\gamma_j}_{<0} = 1$. Thus, there exists at least one index $k \neq i$ such that $\gamma_k < 0$ and $a_{ik} > 0$.
- Assume that $\gamma_i \leq 0$, then we deduce that $\underbrace{\gamma_i}_{\leq 0} + \underbrace{\sum_{j=1,\ j\neq i}^{n} a_{ij}\gamma_j}_{\geq 1} = 1$. Thus, there exists a $\gamma_k > 1$ for some index $k \neq i$. Indeed, since A is nonnegative and strictly diagonally dominant by rows, even if all $\gamma_j = 1$ for $j \neq i$, we would have $\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\gamma_j = \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} < a_{ii} = 1$ which does not satisfy the equation. Similarly, if all $\gamma_j < 1$ for $j \neq i$, we still have $\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\gamma_j < 1$. Hence, there exists at least one index $k \neq i$ such that $\gamma_k > 1$.

To summarize, we have the following implications:

- $\gamma_i > 1 \implies \exists \gamma_k < 0 \quad k \neq i$
- $\gamma_i \leq 0 \implies \exists \gamma_k > 1 \quad k \neq i$, from the first point we deduce that $\exists \gamma_l < 0 \quad l \neq k$.

These implications form a circle. We will now prove that having a $\gamma_i > 1$ is in fact impossible, and therefore $\gamma_i \in (0, 1]$ for $i = 1, \dots, n$.

Let i and k be two indices such that $\gamma_i = \max_j \gamma_j$ and $\gamma_k = \min_j \gamma_j$. Assuming that $\exists \gamma_j > 1$ implies that $\gamma_i = 1 + d$ with $d > 0$ since it is the maximum. Then, from equation (1)

$$-\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\gamma_j = d.$$

Since $\gamma_i > 1$, from our previous findings there exists a $\gamma_j < 0$. Thus, $\gamma_k < 0$ since it is the minimum. Now, let us remark that

$$d = -\sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij}\gamma_j \leq -\gamma_k \sum_{\substack{j=1 \\ j\neq i}}^{n} a_{ij} < -\gamma_k,$$

where again we have used the assumption that A is strictly diagonally dominant by rows. Thus, $\gamma_i < 1 - \gamma_k$. Now considering equation k of the linear system $A\boldsymbol{\gamma} = \mathbf{1}$, we have

$$\gamma_k + \sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}\gamma_j = 1,$$

thus,

$$1 - \gamma_k = \sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj}\gamma_j \leq \gamma_i \sum_{\substack{j=1 \\ j\neq k}}^{n} a_{kj} < \gamma_i$$

and $\gamma_i > 1 - \gamma_k$ which is a contradiction. Hence, $\gamma_i > 1$ is impossible. Since assuming the existence of a $\gamma_k \leq 0$ leads to the existence of a $\gamma_i > 1$, having a $\gamma_k \leq 0$ is also impossible. Therefore, $\gamma_i \in (0, 1]$ for $i = 1, \dots, n$. □

The additional assumptions of nonnegativity with unit diagonal entries are not limiting, as they are satisfied in particular by Wendland C^2, and Gaussian splines. Let us summarize the main results of this section. Provided the following conditions are satisfied:

- C1 Φ_{NM} does not contain a row of zeros,
- C2 Φ_{MM} is strictly diagonally dominant by rows,
- C3 (a) Φ_{MM} is nonnegative with unit diagonal,
  (b) Φ_{NM} is nonnegative,

then the matrix Φ_{MM} is invertible and $\mathbf{g}_\zeta = \Phi_{NM}\Phi_{MM}^{-1}\mathbf{1}_\xi > 0$ component-wise.

2.3. The choice of the radius

We will now discuss how to ensure that the constructed matrix Φ_{MM} is strictly diagonally dominant by rows while restricting the discussion to the compactly supported Wendland C^2 radial basis functions. In [8], different choices of RBF were considered, but the rescaled-localized Wendland C^2 radial basis functions were the most promising because of their good interpolation accuracy while being only locally supported. Moreover, the numerical experiments reported in [8] revealed that the condition number of Φ_{MM} grew very slowly with the size of the interpolation problem. Since matrix-vector products with the interpolation matrix R_{NM} require solving a linear system with Φ_{MM}, a small condition number for this matrix is desirable. We recall that the Wendland C^2 radial basis functions are defined as $\phi(\delta) = (1-\delta)_+^4(1+4\delta)$. Note that this definition suggests $\phi(\delta)$ decays extremely fast as δ grows.

We must investigate how to enforce conditions C1 and C2. In fact, for computational reasons, we will enforce a stronger condition (than C1) on Φ_{NM}: any of its nonzero entries must be safely away from zero. This requirement is set to avoid dividing by a very small number (potentially even smaller than machine precision). However, it was shown in [15] that the entries of $\boldsymbol{\gamma}$ could be arbitrarily close to zero. Thus, the condition on the entries of Φ_{NM} is not enough: we could potentially encounter the situation where the only nonzero in a row of Φ_{NM} multiplies an entry of $\boldsymbol{\gamma} = \Phi_{MM}^{-1}\mathbf{1}_\xi$ which is ϵ away from zero. Fortunately, this situation is unlikely in practice. For any $j = 1, \dots, M$, let r_j denote the radius of the basis function centered at $\boldsymbol{\xi}_j$. We will enforce the following conditions:

$$\exists c \in (0,1): \quad \forall j = 1,\dots,M \ \|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2 \geq c r_j \text{ for any } i \neq j, \tag{2}$$

$$\exists C \in (c,1): \quad \forall i = 1,\dots,N \ \exists j: \ \|\boldsymbol{\zeta}_i - \boldsymbol{\xi}_j\|_2 \leq C r_j. \tag{3}$$

Condition (2) prevents quick loss of diagonal dominance of Φ_{MM}. Indeed, if two distinct points $\boldsymbol{\xi}_i$ and $\boldsymbol{\xi}_j$ are too close to each other (relative to the radius chosen), then $\phi(\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|, r_j) \approx \phi(0, r_j) = 1$ and diagonal dominance will be quickly lost. Condition (2) alone is critical and prevents us from choosing a too large radius. On the other hand, condition (3) avoids having a single nonzero entry in a row of Φ_{NM} which is too close to zero. Condition (3) is equivalent to stating that any point $\boldsymbol{\zeta}_i$ is well within the support of at least one interpolation point $\boldsymbol{\xi}_j$. In other words, it is safely away from the boundary of the support. If this condition is not met, the point $\boldsymbol{\zeta}_i$ will be considered isolated and removed from the set of evaluation points. A feasible situation is illustrated in Fig. 1. From the figure, the number of nonzeros in the jth column of Φ_{MM} and Φ_{NM} can be deduced: they are the number of points $\boldsymbol{\xi}_i$ for $i = 1,\dots,M$ and $\boldsymbol{\zeta}_i$ for $i = 1,\dots,N$, respectively, in the support of $\boldsymbol{\xi}_j$. Both are equal to 3 in Fig. 1. However, the number of nonzeros in each row of Φ_{MM} and Φ_{NM} can only be deduced once all radiuses have been computed. More specifically, the number of nonzeros in the ith row of Φ_{MM} is the number of supports to which point $\boldsymbol{\xi}_i$ belongs. Similarly, the number of nonzeros in the ith row of Φ_{NM} is the number of supports to which point $\boldsymbol{\zeta}_i$ belongs. If the point $\boldsymbol{\zeta}_i$ does not belong to any support, then the ith row of Φ_{NM} will contain only zeros and the rescaling fails. Such a point will also be considered isolated and removed from the set of evaluation points.

Thanks to condition (2), and since ϕ is a decreasing function of $\|\mathbf{x}\|$, we deduce that

$$\phi(\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|, r_j) \leq (1-c)^4(1+4c) \quad i \neq j.$$

Let us assume that point $\boldsymbol{\xi}_i$ belongs to the support of n different radial basis functions (excluding itself). Then, there are n nonzero off-diagonal elements in the ith row of Φ_{MM}. Consequently,

$$\sum_{\substack{j=1 \\ j\neq i}}^{M} \phi(\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|, r_j) \leq n(1-c)^4(1+4c).$$

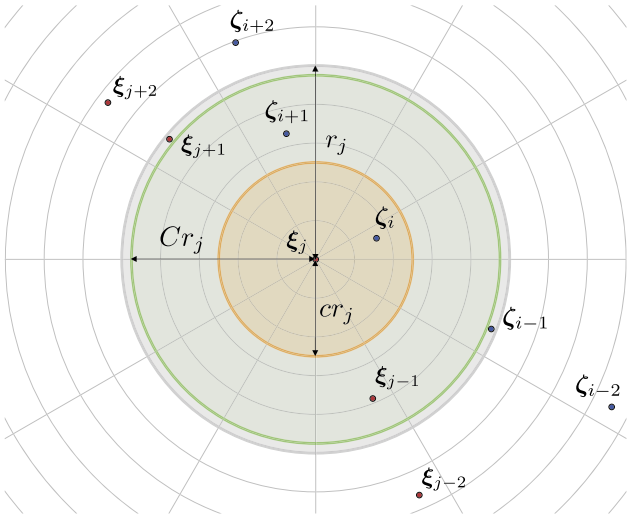

Fig. 1. A feasible radius r_j satisfying condition (2). Point $\boldsymbol{\zeta}_{i-1}$ belongs indeed to the support of $\boldsymbol{\xi}_j$ but $\|\boldsymbol{\zeta}_{i-1} - \boldsymbol{\xi}_j\|_2 > Cr_j$. If $\nexists k: \ \|\boldsymbol{\zeta}_{i-1} - \boldsymbol{\xi}_k\|_2 \leq Cr_k$, then the point $\boldsymbol{\zeta}_{i-1}$ will be considered isolated.

In order to enforce strict diagonal dominance by rows, we set the condition

$$n(1-c)^4(1+4c) < \phi(0, r_i) = 1 \iff n < \frac{1}{(1-c)^4(1+4c)}. \tag{4}$$

Thus, the interpolation point $\boldsymbol{\xi}_i$ must not belong to too many different supports. The number of supports is controlled by the parameter c. In practice, it must be sufficiently greater than 0 but the range of feasible values depends on the dimension of the problem. In order to choose a suitable value of c, we can think of how many nonzeros we should allow in each row of Φ_{MM}. Our choice is guided by the increasing function $f(c) = \frac{1}{(1-c)^4(1+4c)}$. We must allow n to be sufficiently large for feasibility purposes. In fact, $n \geq 2$ is a minimum requirement for 2D problems, based on the application we will consider next. Consequently, c must also be sufficiently large. On the other hand, choosing c too large will much increase the orange area in Fig. 1 and might eventually also lead to an infeasible problem. In all our experiments, we chose $c = 0.5$ and never encountered any problem. This implies $n \leq 5$ and Φ_{MM} will never contain more than 5 off-diagonal nonzeros per row. Thus, choosing the parameter c determines n. The algorithm then proceeds iteratively: each radius r_j is selected such that it satisfies condition (2). This choice is not unique. An initial guess is made for the radius. If it violates condition (2) (the radius is too large), r_j is set to the largest feasible value $\frac{\min_{i\neq j}\|\boldsymbol{\xi}_i - \boldsymbol{\xi}_j\|_2}{c}$. Once all radii are known, the number of nonzero off-diagonal elements in each row of Φ_{MM} is determined. If any of these numbers is greater than n, the matrix Φ_{MM} may not be strictly diagonally dominant. In this case, the parameter c is slightly increased (allowing n to increase as well) and the process is repeated until the number of off-diagonal nonzeros in each row does not exceed n.

Choosing C is less critical and depends only on the function $f(C) = (1-C)^4(1+4C)$. Any value of C for which $f(C)$ is safely away from machine precision is suitable. Thus, C must be sufficiently smaller than 1. In our experiments we chose $C = 0.95$. Consequently, if $\boldsymbol{\zeta}_i$ is not an isolated node, then there exists an index j such that $\phi(\|\boldsymbol{\zeta}_i - \boldsymbol{\xi}_j\|, r_j) \geq 3 \times 10^{-5}$ thanks to inequality (3). The values for the parameters c and C mentioned here are only indicative. In our experiments, good results were obtained over a relatively large range of values.

The discussion held so far shows that the conditions C1 and C2 which are enforced on the matrices Φ_{NM} and Φ_{MM}, respectively, are controlled by the radii of the radial basis functions only. Thus, the matrices Φ_{NM} and Φ_{MM} which are subsequently assembled satisfy the conditions by construction.

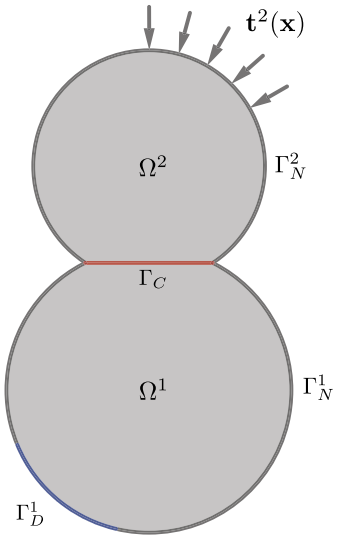

Fig. 2. Contact problem between two deformable bodies.

3. The INTERNODES method for contact problems

The mathematical description of contact problems has been treated in several references. For an in-depth discussion of the functional analysis, we refer the reader to the mathematical literature such as in [16,17]. A discussion related to computational and implementation aspects can be found in [18]. The description we provide in this section is neither complete nor rigorous. Instead, we wish to illustrate the application of the INTERNODES method for the numerical solution of contact problems. The three most popular solution methods in computational contact mechanics are the penalty method, Lagrange multiplier method and augmented Lagrangian method. The method presented in this chapter is a Lagrange multiplier method.

In this section, we will first consider the continuous description of a contact problem between two deformable elastic bodies. We will assume a linear elastic constitutive model in static loading conditions. The following three subsections are a formalization of the work of [7]. The strong form is first stated and the weak form is then established. The finite element discretization is covered and combined with the INTERNODES method: it leads to an algebraic system of equations and the associated matrix will be referred to as the INTERNODES matrix. The nonlinear nature of the contact constraints requires in fact solving a sequence of such linear systems. Their efficient solution will be the focus of the upcoming sections.

3.1. Strong form

The strong (or differential) form of the problem stems from the balance of momentum and Newton's laws of motion. We must define the mathematical properties of the quantities involved. Let $\Omega^k,\ k=1,2$, be an open connected domain of $\mathbb{R}^d,\ d=2,3$, with Lipschitz boundary such that $\Omega^1 \cap \Omega^2 = \emptyset$. Let Γ_D^k, Γ_N^k and Γ_C^k form a partition of the boundary $\partial\Omega^k$. The contact interface Γ_C is common to both bodies. Thus, $\Gamma_C^1 = \Gamma_C^2 = \Gamma_C$. Γ_D^k and Γ_N^k are the Dirichlet and the Neumann external boundaries of Ω^k, respectively. We write $\partial\Omega^k = \overline{\Gamma_D^k} \cup \overline{\Gamma_N^k} \cup \overline{\Gamma_C^k}$ with $\Gamma_D^k \cap \Gamma_N^k = \emptyset$, $\Gamma_D^k \cap \Gamma_C^k = \emptyset$ and $\Gamma_N^k \cap \Gamma_C^k = \emptyset$. An illustration is provided in Fig. 2. In this section, we will make all the necessary assumptions on the regularity of the data for the problem to be well-posed.

Our goal is to approximate the displacement field $\mathbf{u}^k$ of each body $k=1,2$. We will formulate the equilibrium equations for a given body k. The strong form of the problem reads: find $\mathbf{u}^k : \Omega^k \to \mathbb{R}^d$ for $k=1,2$, such that

$$\begin{cases} -\operatorname{div}(\sigma(\mathbf{u}^k)) = \mathbf{f}^k(\mathbf{x}) & \text{in } \Omega^k \quad \text{(a)} \\ \mathbf{u}^k(\mathbf{x}) = \mathbf{g}^k(\mathbf{x}) & \text{on } \Gamma_D^k \quad \text{(b)} \\ \sigma(\mathbf{u}^k)\mathbf{n}^k = \mathbf{t}^k(\mathbf{x}) & \text{on } \Gamma_N^k \quad \text{(c)} \\ \sigma(\mathbf{u}^k)\mathbf{n}^k = \lambda^k(\mathbf{x}) & \text{on } \Gamma_C \quad \text{(d)} \\ \lambda^k(\mathbf{x}) \cdot \mathbf{n}^k \le 0 & \text{on } \Gamma_C \quad \text{(e)} \end{cases} \tag{5}$$

Equation (5a) is the differential equation to be solved, $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is the Cauchy stress tensor, and $\mathbf{f}^k \in [L^2(\Omega^k)]^d$ is the body force acting on the solid (for example the gravity). Equation (5b) prescribes Dirichlet boundary conditions, $\mathbf{g}^k$ is a prescribed displacement field on Γ_D^k. Equation (5c) prescribes Neumann boundary conditions, $\mathbf{t}^k \in [L^2(\Gamma_N^k)]^d$ are surface tractions prescribed on Γ_N^k and $\mathbf{n}^k$ is the unit outward normal vector. On the contact interface Γ_C, the normal vectors point in opposite directions ($\mathbf{n}^2 = -\mathbf{n}^1$). Equation (5d) results from the interaction of the two bodies. λ^k is the stress vector at the contact interface and is unknown. Inequality (5e) is one of the Hertz-Signorini-Moreau conditions [19] which are equivalent to the Karush-Kuhn-Tucker (KKT) conditions in optimization [20,21]. Physically speaking, the stress state must be in compression all along the contact interface. Compression is negative according to the sign convention used in structural mechanics.

These sets of equations must be complemented with another of the Hertz-Signorini-Moreau conditions. It takes the form of a compatibility condition of the displacement fields at the interface and ensures the coupling between the two bodies. Indeed, in the deformed configuration, the current position on the contact interface must be identical. Thus,

$$\mathbf{u}^1 + \mathbf{r}^1 = \mathbf{u}^2 + \mathbf{r}^2 \text{ on } \Gamma_C \tag{6}$$

where $\mathbf{r}^k$ is the position vector in the initial configuration. Furthermore, note that λ^1 and λ^2 are related through Newton's third law by $\lambda^2 = -\lambda^1$. In linear elasticity, the stress tensor σ is a linear function of the infinitesimal strain tensor

$$\epsilon(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T),$$

for some displacement field $\mathbf{u}$, which allows to introduce the constitutive model as

$$\sigma(\mathbf{u}) = C : \epsilon(\mathbf{u}) \tag{7}$$

where C is the linear elastic fourth-order tensor, which can be simplified to

$$C_{ijkl} = \lambda\delta_{ij}\delta_{kl} + \mu(\delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}), \tag{8}$$

for isotropic materials, with $\mu, \lambda \ge 0$ the Lamé constants which are material dependent (see [22] for a comprehensive introduction to constitutive equations). We will restrict the discussion in this paper to materials which follow this law.

3.2. Weak form

The surface tractions are unknown on the boundary Γ_D^k. Hence, let us define $V^k = [H^1(\Omega^k)]^d$ and choose $\mathbf{v}^k \in V_0^k$ where $V_0^k = \{\mathbf{v} \in V^k :\ \mathbf{v}|_{\Gamma_D^k} = \mathbf{0}\}$, for $k=1,2$, where the evaluation on the boundary must be understood in the sense of traces of functions in V^k. Let us furthermore define the space $V_{\Gamma_D}^k = \{\mathbf{v} \in V^k :\ \mathbf{v}|_{\Gamma_D^k} = \mathbf{g}^k\}$ and $\Lambda = [L^2(\Gamma_C)]^d$. The weak form of (5a)–(5d) reads: find $(\mathbf{u}^k, \lambda^k) \in V_{\Gamma_D}^k \times \Lambda$ for $k=1,2$ such that

$$\begin{aligned} &\int_{\Omega^k} \epsilon(\mathbf{u}^k) : C : \epsilon(\mathbf{v}^k)\, \mathrm{d}\Omega - \int_{\Gamma_C} \lambda^k \cdot \mathbf{v}^k\, \mathrm{d}S \\ &= \int_{\Omega^k} \mathbf{f}^k \cdot \mathbf{v}^k\, \mathrm{d}\Omega + \int_{\Gamma_N^k} \mathbf{t}^k \cdot \mathbf{v}^k\, \mathrm{d}S \quad \forall \mathbf{v}^k \in V_0^k. \end{aligned} \tag{9}$$

We define the bilinear forms $a_k : V^k \times V^k \to \mathbb{R}$ and $b_k : V^k \times \Lambda \to \mathbb{R}$

$$a_k(\mathbf{u}^k,\mathbf{v}^k)=\int_{\Omega^k}\epsilon(\mathbf{u}^k):\mathcal{C}:\epsilon(\mathbf{v}^k)\,\mathrm{d}\Omega \qquad \text{and}$$

$$b_k(\mathbf{v}^k,\lambda^k)=-\int_{\Gamma_C}\mathbf{v}^k\cdot\lambda^k\,\mathrm{d}S,$$

and the linear functional $F_k:V^k\to\mathbb{R}$

$$F_k(\mathbf{v}^k)=\int_{\Omega^k}\mathbf{f}^k\cdot\mathbf{v}^k\,\mathrm{d}\Omega+\int_{\Gamma_N^k}\mathbf{t}^k\cdot\mathbf{v}^k\,\mathrm{d}S.$$

Thus, equation (9) may be more compactly written as: find $(\mathbf{u}^k,\lambda^k)\in V^k_{\Gamma_D}\times\Lambda$ for $k=1,2$ such that

$$a_k(\mathbf{u}^k,\mathbf{v}^k)+b_k(\mathbf{v}^k,\lambda^k)=F_k(\mathbf{v}^k)\quad\forall\mathbf{v}^k\in V_0^k.$$

Now, for the compatibility conditions in equation (6), taking the inner product with a test function $\mathbf{w}\in\Lambda$ and integrating over the contact interface leads to

$$\int_{\Gamma_C}\mathbf{u}^1\cdot\mathbf{w}\,\mathrm{d}S-\int_{\Gamma_C}\mathbf{u}^2\cdot\mathbf{w}\,\mathrm{d}S=\int_{\Gamma_C}\mathbf{r}^2\cdot\mathbf{w}\,\mathrm{d}S-\int_{\Gamma_C}\mathbf{r}^1\cdot\mathbf{w}\,\mathrm{d}S,$$

that may be compactly written as $(\mathbf{u}^1-\mathbf{u}^2,\mathbf{w})_{L^2(\Gamma_C)}=(\mathbf{r}^2-\mathbf{r}^1,\mathbf{w})_{L^2(\Gamma_C)}$ where $(\cdot,\cdot)_{L^2(\Gamma_C)}$ denotes the inner product in $L^2(\Gamma_C)$. To account for the Dirichlet boundary conditions, we write $\mathbf{u}^k=\mathbf{u}_0^k+\mathbf{s}^k$ where $\mathbf{u}_0^k\in V_0^k$ and $\mathbf{s}^k\in V^k$ is such that $\mathbf{s}^k|_{\Gamma_D^k}=\mathbf{g}^k$ in the sense of the trace. We may directly replace $\mathbf{u}^k$ with $\mathbf{u}_0^k+\mathbf{s}^k$ and all the previous equations can now be combined in a single system as: look for $(\mathbf{u}_0^k,\lambda^k)\in V_0^k\times\Lambda$ for $k=1,2$ such that

$$\begin{aligned}
&a_1(\mathbf{u}_0^1,\mathbf{v}^1)+b_1(\mathbf{v}^1,\lambda^1)=F_1(\mathbf{v}^1)-a_1(\mathbf{s}^1,\mathbf{v}^1)\quad\forall\mathbf{v}^1\in V_0^1\\
&a_2(\mathbf{u}_0^2,\mathbf{v}^2)+b_2(\mathbf{v}^2,\lambda^2)=F_2(\mathbf{v}^2)-a_2(\mathbf{s}^2,\mathbf{v}^2)\quad\forall\mathbf{v}^2\in V_0^2\\
&(\mathbf{u}_0^1-\mathbf{u}_0^2,\mathbf{w})_{L^2(\Gamma_C)}=(\mathbf{r}^2-\mathbf{r}^1,\mathbf{w})_{L^2(\Gamma_C)}\quad\forall\mathbf{w}\in\Lambda\\
&\lambda^2=-\lambda^1.
\end{aligned}\tag{10}$$

Equations (10) form a saddle point type system which is typical from constrained optimization problems. The system is so named because the solution is a saddle point of a Lagrangian function. These equations are the first order conditions for a stationary point of the Lagrangian function. See for instance [23] for the theory of saddle point problems and their well-posedness. We will now derive their discrete counterpart obtained through the finite element method.

3.3. Finite element approximation

We will approximate problem (10) by the Galerkin approach and more precisely by using the finite element method. For this purpose we introduce a triangulation τ_h^k of the domain Ω^k for $k=1,2$ where $\tau_h^k=\bigcup_{i=1}^{N_k}K_i^k$ with K_i^k being the finite elements and N_k the number of elements used to approximate the domain Ω^k. The triangulations are typically nonconforming in the vicinity of the discrete contact interface. The nonconformity is essentially geometric. There might exist small gaps or overlaps. However, τ_h^1 and τ_h^2 taken alone are conforming. We are seeking an approximation $\mathbf{u}_{0,h}^k$ of $\mathbf{u}_0^k$ with $\mathbf{u}_{0,h}^k\in V_{0,h}^k\subset V_0^k$. Similarly, we look for an approximation $\lambda_h^k\in\Lambda_h^k\subset\Lambda^k$. Note that in the discrete problem, the interface spaces Λ_h^1 and Λ_h^2 must be distinguished. Moreover, we must choose in which of these two spaces the test function $\mathbf{w}_h$ belongs. This choice amounts to deciding which body should be the master (using the terminology of mortar methods). For the discrete problem, we introduce some finite element spaces. Let $X_{h,r}^k$ be the space of piecewise polynomials of degree r in each of the d components with $d=2,3$ in $\overline{\Omega_h^k}$, the approximation of $\overline{\Omega^k}$, such that

$$X_{h,r}^k=\{\mathbf{v}_h\in[C^0(\overline{\Omega_h^k})]^d:\ \mathbf{v}_h|_K\in[\mathbb{P}_r]^d\quad K\in\tau_h^k\}\quad k=1,2.$$

Its subsets are

$$V_{h,0}^k=\{\mathbf{v}_h\in X_{h,r}^k:\ \mathbf{v}_h|_{\Gamma_D^k}=\mathbf{0}\},\qquad V_{h,\Gamma_D}^k=\{\mathbf{v}_h\in X_{h,r}^k:\ \mathbf{v}_h|_{\Gamma_D^k}=\mathbf{g}^k\}.$$

We must now introduce two intergrid transfer operators, $\Pi_{12}:\Lambda_h^2\to\Lambda_h^1$ and $\Pi_{21}:\Lambda_h^1\to\Lambda_h^2$. The distinguishing feature of the INTERNODES method is that they are interpolation operators (see [4,24]). In particular, due to the geometric nonconformities encountered in our problem, they will be based on radial basis function interpolation. The finite element approximation of (10), in which the coupling is achieved by INTERNODES, reads: find $(\mathbf{u}_{0,h}^k,\lambda_h^k)\in V_{h,0}^k\times\Lambda_h^k$ for $k=1,2$ such that

$$\begin{aligned}
&a_1(\mathbf{u}_{0,h}^1,\mathbf{v}_h^1)+b_1(\mathbf{v}_h^1,\lambda_h^1)=F_1(\mathbf{v}_h^1)-a_1(\mathbf{s}_h^1,\mathbf{v}_h^1)\quad\forall\mathbf{v}_h^1\in V_{0,h}^1\\
&a_2(\mathbf{u}_{0,h}^2,\mathbf{v}_h^2)+b_2(\mathbf{v}_h^2,\lambda_h^2)=F_2(\mathbf{v}_h^2)-a_2(\mathbf{s}_h^2,\mathbf{v}_h^2)\quad\forall\mathbf{v}_h^2\in V_{0,h}^2\\
&\left(\mathbf{u}_{0,h}^1|_{\Gamma_C^1}-\Pi_{12}(\mathbf{u}_{0,h}^2|_{\Gamma_C^2}),\mathbf{w}_h\right)_{L^2(\Gamma_C^1)}\\
&\quad=\left(\Pi_{12}(\mathbf{r}_h^2|_{\Gamma_C^2})-\mathbf{r}_h^1|_{\Gamma_C^1},\mathbf{w}_h\right)_{L^2(\Gamma_C^1)}\quad\forall\mathbf{w}_h\in\Lambda_h^1\\
&\lambda_h^2=-\Pi_{21}\lambda_h^1.
\end{aligned}\tag{11}$$

3.4. Algebraic system of equations

Since any $\mathbf{v}_h^k$ for $k=1,2$ and $\mathbf{w}_h$ can be expressed as a linear combination of the basis functions of the respective finite dimensional spaces, it is enough to enforce all the equations of (11) for all basis functions of the appropriate spaces. Moreover, $\mathbf{u}_{0,h}^k$ and λ_h^k can be expanded with respect to the Lagrange basis of the respective space as

$$\mathbf{u}_{0,h}^k(\mathbf{x})=\sum_j u_j^k\boldsymbol{\psi}_j^k(\mathbf{x}),\quad \lambda_h^k(\mathbf{x})=\sum_j\lambda_j^k\boldsymbol{\phi}_j^k(\mathbf{x}),$$

where $\{\boldsymbol{\psi}_j^k\}$ and $\{\boldsymbol{\phi}_j^k\}$ are the basis of $V_{0,h}^k$ and Λ_h^k, respectively. Notice that $\{\boldsymbol{\phi}_j^k\}$ are the restrictions to Γ_C^k of the basis functions $\{\boldsymbol{\psi}_j^k\}$. Furthermore, as we have already noted, the coefficients λ_j^k for $k=1,2$ are related. In the discrete setting, this relation may be conveniently expressed using the intergrid transfer operators. These operators act separately on each component of a vector valued function. Therefore, the presentation can be restricted to a single component. By a slight abuse of notation, we will also denote Π_{21} the interpolation operator acting on individual components. Thus, for any component j of the vector functions we have:

$$\begin{aligned}
\lambda_{h,j}^2(\mathbf{x})&=-\Pi_{21}\lambda_{h,j}^1(\mathbf{x})\\
&=-\sum_l(\Pi_{21}\lambda_{h,j}^1)(\mathbf{x}_l^2)\phi_l^2(\mathbf{x})=-\sum_l\sum_i\lambda_{i,j}^1(\Pi_{21}\phi_i^1)(\mathbf{x}_l^2)\phi_l^2(\mathbf{x}),
\end{aligned}$$

where l and i denote summation indices taken over the interface nodes of body 2 and 1, respectively, and $\mathbf{x}_l^2$ are the interface nodes of body 2. In the INTERNODES method, we need small interface mass matrices of the type

$$(M_k)_{ij}=(\boldsymbol{\phi}_j^k,\boldsymbol{\phi}_i^k)_{L^2(\Gamma_C^k)},\qquad k=1,2,\tag{12}$$

where the indices i and j span the range of degrees of freedom on the interface Γ_C^k. After a suitable ordering of the unknowns, the interpolation matrices R_{12} and R_{21} associated with the interpolation operators Π_{12} and Π_{21}, respectively, are block diagonal with identical blocks defined as

$$(R_{12}^b)_{ij}=(\Pi_{12}\phi_j^2)(\mathbf{x}_i^1),\quad (R_{21}^b)_{ij}=(\Pi_{21}\phi_j^1)(\mathbf{x}_i^2),\quad b=1,\dots,d,$$

for appropriate indices i and j taken over the interface nodes. The algebraic system coupled by INTERNODES reads

$$\begin{pmatrix}
K_{\Omega_1\Omega_1} & K_{\Omega_1\Gamma_1} & 0 & 0 & 0\\
K_{\Gamma_1\Omega_1} & K_{\Gamma_1\Gamma_1} & 0 & 0 & -M_1\\
0 & 0 & K_{\Omega_2\Omega_2} & K_{\Omega_2\Gamma_2} & 0\\
0 & 0 & K_{\Gamma_2\Omega_2} & K_{\Gamma_2\Gamma_2} & M_2R_{21}\\
0 & M_1 & 0 & -M_1R_{12} & 0
\end{pmatrix}
\begin{pmatrix}\mathbf{u}_{\Omega_1}\\ \mathbf{u}_{\Gamma_1}\\ \mathbf{u}_{\Omega_2}\\ \mathbf{u}_{\Gamma_2}\\ \lambda\end{pmatrix}=
\begin{pmatrix}\mathbf{f}_{\Omega_1}\\ \mathbf{f}_{\Gamma_1}\\ \mathbf{f}_{\Omega_2}\\ \mathbf{f}_{\Gamma_2}\\ \mathbf{s}\end{pmatrix},$$

where the unknowns are denoted with a subscript Γ_k if they belong to the interface and with Ω_k otherwise, and $k = 1, 2$ refers to the subdomains. The matrices $K_{\Omega_k\Omega_k}$, $K_{\Omega_k\Gamma_k}$, $K_{\Gamma_k\Omega_k}$, and $K_{\Gamma_k\Gamma_k}$, $k = 1, 2$, denote the stiffness matrices for each subdomain, split into their interface and non-interface parts. The vector $\mathbf{s}$ is given by $\mathbf{s} = M_1(R_{12}\mathbf{r}_{\Gamma_2} - \mathbf{r}_{\Gamma_1})$ and R_{12} and R_{21} are the interpolation matrices introduced in the previous section. The matrix M_1 can be eliminated from the last block row as it is symmetric positive definite (and therefore invertible). It leads to solving the linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{pmatrix} K_{\Omega_1\Omega_1} & K_{\Omega_1\Gamma_1} & 0 & 0 & 0 \\ K_{\Gamma_1\Omega_1} & K_{\Gamma_1\Gamma_1} & 0 & 0 & -M_1 \\ 0 & 0 & K_{\Omega_2\Omega_2} & K_{\Omega_2\Gamma_2} & 0 \\ 0 & 0 & K_{\Gamma_2\Omega_2} & K_{\Gamma_2\Gamma_2} & M_2R_{21} \\ 0 & I & 0 & -R_{12} & 0 \end{pmatrix},$$

$$\mathbf{x} = \begin{pmatrix} \mathbf{u}_{\Omega_1} \\ \mathbf{u}_{\Gamma_1} \\ \mathbf{u}_{\Omega_2} \\ \mathbf{u}_{\Gamma_2} \\ \lambda \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{f}_{\Omega_1} \\ \mathbf{f}_{\Gamma_1} \\ \mathbf{f}_{\Omega_2} \\ \mathbf{f}_{\Gamma_2} \\ \mathbf{d} \end{pmatrix},$$

and $\mathbf{d} = R_{12}\mathbf{r}_{\Gamma_2} - \mathbf{r}_{\Gamma_1}$. The second formulation is slightly more appealing from a computational point of view as it avoids unnecessary matrix-vector multiplications with the matrix M_1 when solving linear systems iteratively. Secondly, the vector $\mathbf{d}$ we recover has a direct physical interpretation: it measures the nodal gap in the initial configuration. Although we eventually chose the second formulation, the preconditioning strategies we will investigate in the next sections can be easily adjusted for the first formulation. However, regardless of which formulation is used, one should notice immediately that the sign of the Lagrange multipliers has not been enforced anywhere. Inequality constraints are a common source of nonlinearity in optimization and do not spare contact mechanics [16–18]. In practice, the problem is first relaxed by dropping the inequality constraints on the Lagrange multipliers, hence recovering a linear problem. On the discrete level, it translates to solving a linear system. Once this linear system is solved, the inequality constraints are verified. If any of the constraints are violated, the active set keeping track of the contact interface is updated and the related blocks in the INTERNODES matrix and the right-hand side are updated. This iterative procedure leads to solving a sequence of linear systems $A^{(n)}\mathbf{x}^{(n)} = \mathbf{b}^{(n)}$ for $n = 0, 1, \ldots$ until the constraints are finally satisfied [15,7]. Techniques for solving efficiently this sequence of linear systems will be discussed in Section 5. Before that, some of the properties of the matrices involved will be discussed in the next section.

4. The INTERNODES matrix: its structure and conditioning

As we have seen, the continuous problem already has a saddle point structure which we recover on the discrete level. It seems therefore natural to preserve the structure from which the linear system stems. It leads to considering a 2×2 block structure commonly known as a saddle point system in the literature. Thus, we will most frequently work with the submatrices and subvectors of this linear system defined as follows:

$$K = \begin{pmatrix} K_{\Omega_1\Omega_1} & K_{\Omega_1\Gamma_1} & 0 & 0 \\ K_{\Gamma_1\Omega_1} & K_{\Gamma_1\Gamma_1} & 0 & 0 \\ 0 & 0 & K_{\Omega_2\Omega_2} & K_{\Omega_2\Gamma_2} \\ 0 & 0 & K_{\Gamma_2\Omega_2} & K_{\Gamma_2\Gamma_2} \end{pmatrix} \quad \mathbf{u} = \begin{pmatrix} \mathbf{u}_{\Omega_1} \\ \mathbf{u}_{\Gamma_1} \\ \mathbf{u}_{\Omega_2} \\ \mathbf{u}_{\Gamma_2} \end{pmatrix} \quad \mathbf{f} = \begin{pmatrix} \mathbf{f}_{\Omega_1} \\ \mathbf{f}_{\Gamma_1} \\ \mathbf{f}_{\Omega_2} \\ \mathbf{f}_{\Gamma_2} \end{pmatrix}$$

$$B = \begin{pmatrix} 0 \\ -M_1 \\ 0 \\ M_2R_{21} \end{pmatrix} \quad \tilde{B} = \begin{pmatrix} 0 & I & 0 & -R_{12} \end{pmatrix}$$

and we consider the linear system

$$\underbrace{\begin{pmatrix} K & B \\ \tilde{B} & 0 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \mathbf{u} \\ \lambda \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} \mathbf{f} \\ \mathbf{d} \end{pmatrix}}_{\mathbf{b}}. \tag{13}$$

Let us denote n the size of the stiffness matrix K which represents the bulk of the matrix and m the number of degrees of freedom on the interface of body 1, which is the number of lines of $\tilde{B}$ and the number of columns of B. Thus, $K \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $\tilde{B} \in \mathbb{R}^{m\times n}$ with $n \gg m$ and the vectors $\mathbf{u}, \mathbf{f} \in \mathbb{R}^n$ and $\lambda, \mathbf{d} \in \mathbb{R}^m$. The matrix A will be referred to as the INTERNODES matrix.

It is important to note that contrary to single-body elasticity problems, the stiffness matrix K is only positive semidefinite in general and we denote $s = \dim\ker(K)$, the nullity of K. In the context of mechanics, displacement fields associated to non-trivial vectors in the kernel of the matrix are combinations of translations and rotations. They are called rigid modes, as they do not generate any deformation of the body. Consequently, the kernel of the stiffness matrix is a small dimensional subspace with $s = 3$ in 2D (2 translations, 1 rotation) and $s = 6$ in 3D (3 translations, 3 rotations) at most [25]. More precisely, the stiffness matrix is singular if not all rigid body motions are blocked simultaneously for both bodies. A particular case is when Dirichlet boundary conditions are not prescribed on one of the bodies. More generally, if Dirichlet boundary conditions are allowed to be prescribed on separate components of the displacement vector (as is customary in structural mechanics), both blocks of the stiffness matrix could be singular simultaneously. When investigating computational methods, the structure of K will be entirely ignored as it depends on the numbering of the nodes in the mesh. We must only bear in mind that it is symmetric positive semidefinite.

The matrices B and $\tilde{B}$ are coupling matrices linking the degrees of freedom on the interface. The iterative nature of the contact algorithm means that these matrices will change from one iteration to the next. Clearly the matrices B and $\tilde{B}$ are very sparse. However, they contain small dense submatrices. It is easy to verify that both matrices have full rank m. Indeed, the matrix $\tilde{B}$ contains the identity matrix in one of its blocks whereas the matrix B contains the interface mass matrix of body 1 which, from finite element theory, is known to be symmetric positive definite.

Let us now return to the INTERNODES linear system (13), that is a nonsymmetric saddle point system. Such systems arise in an impressively large collection of scientific disciplines including fluid dynamics [26–28], optimization [29], electromagnetism [30–32] and contact mechanics [33,34,3] to name just a few. Due to the numerous underlying applications, they have been extensively studied by the scientific community and there exists abundant literature on the topic. The survey paper [35] is a well-established reference in the field and highlights the variety of solution methods proposed. We will later rely heavily on it for the design of preconditioners.

Verifying the well-posedness of the discrete finite-dimensional problem amounts to verifying the invertibility of the INTERNODES matrix. The invertibility conditions were recalled and verified in [15]. The verifications revealed that the well-posedness of the problem depends on the boundary conditions prescribed.

Our eventual goal being to efficiently solve linear system (13), information about the conditioning of the INTERNODES matrix is particularly relevant. The next theorem provides an insightful lower bound on the spectral condition number of the INTERNODES matrix. The interested reader is referred to the proof in Appendix A.

Theorem 4.1. *The spectral condition number of the matrix A in (13) satisfies the inequality*

$$\kappa(A) \geq \max\left\{ \kappa_B(K)\max\{1, \frac{\|B\|_2}{\sigma_{1,B}}, \frac{\|\tilde{B}\|_2}{\sigma_{1,B}}\},\ \kappa_{\tilde{B}}(K)\max\{1, \frac{\|B\|_2}{\sigma_{1,\tilde{B}}}, \frac{\|\tilde{B}\|_2}{\sigma_{1,\tilde{B}}}\} \right\}$$

where

$$\sigma_{1,B} = \max_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(B^T)}} \|K\mathbf{x}\|_2 \quad \sigma_{n,B} = \min_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(B^T)}} \|K\mathbf{x}\|_2 \quad \kappa_B(K) = \frac{\sigma_{1,B}}{\sigma_{n,B}}$$

$$\sigma_{1,\tilde{B}} = \max_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(\tilde{B})}} \|K\mathbf{x}\|_2 \quad \sigma_{n,\tilde{B}} = \min_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(\tilde{B})}} \|K\mathbf{x}\|_2 \quad \kappa_{\tilde{B}}(K) = \frac{\sigma_{1,\tilde{B}}}{\sigma_{n,\tilde{B}}}.$$

Several important observations stem from this theorem. First of all, it was shown in [36], Proposition 2.1 that for A to be invertible, we have necessarily $\ker(K)\cap\ker(\tilde{B}) = \{\mathbf{0}\}$ and $\ker(K)\cap\ker(B^T) = \{\mathbf{0}\}$. Thus, neither $\sigma_{n,\tilde{B}}$, nor $\sigma_{n,B}$ can be zero. However, the closest these subspaces are in some sense, the smaller $\sigma_{n,\tilde{B}}$ and $\sigma_{n,B}$ could be and thus the larger the condition number of A. Experimentally, we found out that $\kappa_B(K)$ and $\kappa_{\tilde{B}}(K)$ behaved like $O(h^{-2})$ for our application, where h is the mesh size. According to the theorem, this condition number is amplified if $\|B\|_2$ or $\|\tilde{B}\|_2$ are large. We indeed experimentally encountered this situation if the quality of the interpolation was not good enough. This finding is a clear motivation for preconditioning: not only does the condition number grow like h^{-2} but it may even be amplified if the interpolation is defective.

5. Preconditioning techniques

We now turn to the iterative resolution of the INTERNODES system (13). Experimentally, the iterative condition number of the INTERNODES matrix (that is $\mathcal{K}(A) = \max_i|\lambda_i(A)|/\min_i|\lambda_i(A)|$, where $\lambda_i(A)$ are the eigenvalues of A) does not behave very differently from the one for the standard Poisson problem. The usual h^{-2} growth of the iterative condition number with the mesh size h was experienced even for very different applications [4,7]. This property depends on the differential operator and can be expected from second order unbounded operators. Although the iterative condition number alone cannot fully describe the behavior of iterative methods in the nonnormal case, we can still expect the number of iterations to increase with the condition number. Preconditioning techniques are used to keep a small iteration count despite the growing size of the problem. In this section, a preconditioner for the INTERNODES matrix A is proposed and theoretical properties related to the eigenvalues of the preconditioned matrix are proved. The quality of the preconditioner will be shown to depend on a parameter which must be chosen suitably, according to some theoretical results presented thereafter. The resolution of linear systems with this preconditioner is then described with an algorithm.

5.1. Rescaling

The first issue we are facing for the iterative resolution of the INTERNODES system (13) is linked to the different physical nature of the unknowns in the solution vector $\mathbf{x}$. It combines the displacement vector $\mathbf{u}$ and the Lagrange multipliers λ. These quantities may be numerically different by several orders of magnitude. This difference carries over to the different blocks of the INTERNODES matrix. The entries of the stiffness matrix K are extremely large in comparison to those of B and $\tilde{B}$ and this difference is responsible for much of the ill-conditioning of the matrix. We must therefore proceed with a rescaling. In effect, it will lead to a solution vector $\mathbf{x}$ with all entries having the same units. A simple and yet very efficient rescaling is given by

$$\begin{pmatrix} \zeta^{-1}K & B \\ \tilde{B} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \zeta^{-1}\lambda \end{pmatrix} = \begin{pmatrix} \zeta^{-1}\mathbf{f} \\ \mathbf{d} \end{pmatrix}.$$

It simply consists in dividing the entire first block row by a scaling parameter $\zeta > 0$ and performing a change of variables. We redefine $K \leftarrow \zeta^{-1}K$, $\lambda \leftarrow \zeta^{-1}\lambda$ and $\mathbf{f} \leftarrow \zeta^{-1}\mathbf{f}$. The scaling parameter ζ must be chosen according to material properties. The most prominent material parameter, which generates much of the ill-conditioning is the elastic modulus E, which suggests to set $\zeta = E$ in case of a homogeneous material.

In the following, we will refer to the INTERNODES matrix as being the rescaled matrix. Now, the residual norm for the rescaled system $\|\mathbf{r}_r\|_2$ behaves almost like a shift of the unpreconditioned residual norm. We have

$$\|\mathbf{r}\|_2^2 = \underbrace{\|\mathbf{f} - (K\mathbf{u} + B\lambda)\|_2^2}_{\|\mathbf{r}_1\|_2^2} + \underbrace{\|\mathbf{d} - \tilde{B}\mathbf{u}\|_2^2}_{\|\mathbf{r}_2\|_2^2},$$

$$\|\mathbf{r}_r\|_2^2 = \underbrace{\frac{1}{\zeta^2}\|\mathbf{f} - (K\mathbf{u} + B\lambda)\|_2^2}_{\|\mathbf{r}_{1,r}\|_2^2} + \underbrace{\|\mathbf{d} - \tilde{B}\mathbf{u}\|_2^2}_{\|\mathbf{r}_{2,r}\|_2^2}$$

with $\|\mathbf{r}_1\|_2 \gg \|\mathbf{r}_2\|_2$ but $\|\mathbf{r}_{1,r}\|_2 \approx \|\mathbf{r}_{2,r}\|_2$. The components of the residual vector for the rescaled system have the same units. It can be understood as a form of normalization designed to wipe out the effect of material parameters. It allows us to conveniently choose a tolerance for the stopping criterion of an iterative scheme independently of material parameters.

5.2. Spectral properties of the preconditioned matrix

The abundant literature on saddle point systems (see for instance [35–37,27]) provides us with several possibilities for preconditioning the INTERNODES matrix. Let us recall that $K \in \mathbb{R}^{n\times n}$ is symmetric positive semidefinite and $B \in \mathbb{R}^{n\times m}$ and $\tilde{B} \in \mathbb{R}^{m\times n}$ with $\operatorname{rank}(B) = \operatorname{rank}(\tilde{B}) = m$. Special attention must be paid to the properties of the different blocks of the system. In our case, we are seeking a preconditioner for a saddle point system with a singular $(1,1)$ block. For this reason, many preconditioning strategies based on Schur complements (requiring an invertible $(1,1)$ block) do not apply. On the contrary, techniques based on augmenting the $(1,1)$ block by adding a low rank term are most appropriate when facing a singular $(1,1)$ block. These techniques lead to the large class of augmented Lagrangian preconditioners. The preconditioner we propose is a slight modification of an augmented Lagrangian preconditioner proposed in [38] and based on earlier results from [36]. In [38], the author proposed a preconditioner of the form

$$M = \begin{pmatrix} K + BW^{-1}\tilde{B} & kB \\ 0 & -W \end{pmatrix}$$

where $k \in \mathbb{R}$ is a scalar parameter, $W \in \mathbb{R}^{m\times m}$ is an invertible weight matrix and we assume $K + BW^{-1}\tilde{B}$ is invertible. In [38], the author proved theoretical properties related to the clustering of the eigenvalues of the preconditioned system. Here, we shall consider a generalization of the preconditioner of [38] that consists in adding scalar parameters $\alpha, \beta, \gamma \in \mathbb{R}\setminus\{0\}$. We therefore consider

$$M = \begin{pmatrix} K + \alpha BW^{-1}\tilde{B} & -\beta\gamma^{-1}B \\ 0 & -\gamma^{-1}W \end{pmatrix}. \tag{14}$$

We will subsequently discuss suitable choices for the parameters α, β and γ as well as for the matrix W. The left preconditioned system reads $M^{-1}A\mathbf{x} = M^{-1}\mathbf{b}$. Studying the eigenvalues of the preconditioned matrix $M^{-1}A$ amounts to studying the eigenvalues of the generalized eigenvalue problem $A\mathbf{v} = \lambda M\mathbf{v}$, that is

$$\begin{pmatrix} K & B \\ \tilde{B} & 0 \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix} = \lambda\begin{pmatrix} K + \alpha BW^{-1}\tilde{B} & -\beta\gamma^{-1}B \\ 0 & -\gamma^{-1}W \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix}. \tag{15}$$

The second equation in (15) yields $\mathbf{p} = -\gamma\lambda^{-1}W^{-1}\tilde{B}\mathbf{u}$. (Obviously, $\lambda \neq 0$, otherwise the preconditioned matrix $M^{-1}A$ would be singular.) After substituting back in the first equation and regrouping the terms we obtain

$$\lambda(\lambda - 1)K\mathbf{u} + (\alpha\lambda^2 + \beta\lambda + \gamma)BW^{-1}\tilde{B}\mathbf{u} = \mathbf{0}. \tag{16}$$

Three different cases must be analyzed.

- If $\mathbf{u} \in \ker(\tilde{B})$, then equation (16) reduces to $\lambda(\lambda-1)K\mathbf{u} = \mathbf{0}$. Since $K\mathbf{u} \neq \mathbf{0}$, then $\lambda = 1$ is an eigenvalue of algebraic multiplicity $\dim\ker(\tilde{B}) = n - m$.
- If $\mathbf{u} \in \ker(K)$, equation (16) reduces to $(\alpha\lambda^2 + \beta\lambda + \gamma)BW^{-1}\tilde{B}\mathbf{u} = \mathbf{0}$. Since $BW^{-1}\tilde{B}\mathbf{u} \neq \mathbf{0}$, then $\alpha\lambda^2 + \beta\lambda + \gamma = 0$, which yields

  $$\lambda_{1,2} = \frac{-\beta \pm \sqrt{\beta^2 - 4\alpha\gamma}}{2\alpha}.$$

  These eigenvalues have algebraic multiplicity s each. We must still identify $n + m - (n - m + 2s) = 2(m - s)$ eigenvalues. Note that if we set $\beta = -2\alpha$ and $\gamma = \alpha$, then we again obtain $\lambda_{1,2} = 1$ which is very favorable, as we already have a cluster of eigenvalues located at 1.
- If $\mathbf{u} \notin \{\ker(K) \cup \ker(\tilde{B})\}$, then both terms of equation (16) must be considered. After rearranging the terms, we obtain

  $$K\mathbf{u} = \frac{\alpha\lambda^2 + \beta\lambda + \gamma}{\lambda(1-\lambda)} BW^{-1}\tilde{B}\mathbf{u} = \mu BW^{-1}\tilde{B}\mathbf{u},$$

  having denoted $\mu = \frac{\alpha\lambda^2+\beta\lambda+\gamma}{\lambda(1-\lambda)}$. This is again a generalized eigenvalue problem. We can express λ as a function of μ since

  $$\hat{\lambda}_{1,2} = \frac{\mu - \beta \pm \sqrt{(\beta-\mu)^2 - 4\gamma(\alpha+\mu)}}{2(\alpha+\mu)}. \tag{17}$$

  In fact, if we set $\beta = -2\alpha$ and $\gamma = \alpha$, the expression simplifies considerably as $\Delta = (\beta-\mu)^2 - 4\gamma(\alpha+\mu) = \mu^2$. Therefore, the remaining eigenvalues are

  $$\hat{\lambda}_1 = 1 \text{ and } \hat{\lambda}_2 = \frac{\alpha}{\alpha+\mu}.$$

  Clearly, $\lim_{\alpha\to\infty} \hat{\lambda}_2 = 1$. However, in practice, any α such that $|\alpha| \gg |\mu|$ will already lead to an increasingly good eigenvalue clustering. As a matter of fact, even if the generalized eigenvalues μ are in general complex, when $|\alpha| \gg |\mu|$, the imaginary part of $\hat{\lambda}_2$ is very small and this guarantees the eigenvalue clustering around 1, providing the good conditioning of the preconditioned matrix.

In summary, provided α and W are chosen suitably, the eigenvalues of the preconditioned matrix are $\lambda = 1$ of multiplicity $n - m$ if $\mathbf{u} \in \ker(\tilde{B})$, $\lambda = 1$ of multiplicity $2s$ if $\mathbf{u} \in \ker(K)$, $\lambda = 1$ of multiplicity $(m - s)$ and $\lambda \approx 1$ of multiplicity $(m - s)$ if $\mathbf{u} \in \mathbb{C}^n \setminus \{\ker(K) \cup \ker(\tilde{B})\}$. Naturally, the values μ that are the generalized eigenvalues of

$$K\mathbf{u} = \mu C\mathbf{u}, \quad \text{with } C = BW^{-1}\tilde{B} \tag{18}$$

depend on W. We will therefore proceed in two steps: by first choosing W and characterizing μ and then choosing α suitably. Although the preconditioner defined in (14) is a generalization of the preconditioner proposed in [38], with the choice $\beta = -2\alpha$ and $\gamma = \alpha$, our preconditioner is equivalent to the original preconditioner proposed in [38] after setting $k = 2$ and replacing W by $\alpha^{-1}W$. The advantage of our preconditioner consists in introducing a new weighting parameter α that helps us in clustering the generalized eigenvalues λ of (15). In [38], $\hat{\lambda}_2 = (1+\mu)^{-1}$ and W must be chosen such that $|\mu|$ is small. In our case, we highlight the artificial weighting parameter α while considering a more natural choice for W which will control $|\mu|$. This setup will ease the presentation in the upcoming sections. The main contribution of our work is not in proposing an entirely new preconditioner but rather making better use of an existing one when it is applied to solve the INTERNODES system.

The analysis above revealed that the choices $\beta = -2\alpha$ and $\gamma = \alpha$ are quite advantageous. Moreover, we choose $W = M_1$, the mass matrix on the interface Γ_C^1 defined in (12). Thus, our preconditioner reads

$$M = \begin{pmatrix} K + \alpha B M_1^{-1}\tilde{B} & -2B \\ 0 & \alpha^{-1}M_1^{-1} \end{pmatrix}. \tag{19}$$

In the next subsections we are going to bound the generalized eigenvalues μ, this will help us in setting the parameter α to design the most efficient preconditioner M for the INTERNODES matrix A.

5.3. Sign of the generalized eigenvalues μ

There are only $2(m-s)$ non-trivial eigenvalues μ of problem (18) associated to eigenvectors in the subspace $\mathcal{U} = \mathbb{C}^n \setminus \{\ker(K) \cup \ker(\tilde{B})\}$. All the other eigenvalues μ are either zero or infinity and lead to eigenvalues $\lambda = 1$. The first issue we must resolve is related to the sign of the real and/or imaginary parts of μ. This information is valuable to avoid some unfortunate situations. For example, assuming μ is real and α is chosen such that $|\alpha| \approx |\mu|$ but $\text{sign}(\alpha) = -\text{sign}(\mu)$, then, computing $\alpha + \mu$ would lead to cancellation and this quantity could get dangerously close to zero. Since it appears at the denominator in equation (17), the eigenvalues of the preconditioned matrix could increase tremendously. Numerical experiments confirmed that such an event could have disastrous consequences for the preconditioner. We could safely avoid this situation by knowing the sign of μ. Let us consider the generalized eigenvalue problem

$$K\mathbf{u} = \mu C\mathbf{u} \quad \mathbf{u} \in \mathcal{U} \tag{20}$$

with $C = BM_1^{-1}\tilde{B}$. We define the generalized Rayleigh quotient

$$q(\mathbf{u}) = \frac{\mathbf{u}^* K\mathbf{u}}{\mathbf{u}^* C\mathbf{u}} \quad \mathbf{u} \in \mathcal{S} = \mathbb{C}^n \setminus \{\ker(K) \cup \ker(B^T) \cup \ker(\tilde{B})\}.$$

The subspace over which $\mathbf{u}$ is taken must be further restricted with respect to $\mathcal{U}$ when defining the Rayleigh quotient, indeed, if $\mathbf{u} \in \mathbb{R}^n$ and $\mathbf{u} \in \ker(B^T)$, then the denominator vanishes. We then consider the field of values

$$\mathcal{F} = \left\{ q(\mathbf{u}) : \mathbf{u} \in \mathcal{S} \right\}.$$

Clearly, the numerator of $q(\mathbf{u})$ is positive for all $\mathbf{u} \in \mathcal{S}$. Only the denominator must be analyzed. The matrix C is expressed as $C = -U_1 U_2^T$, where we have defined

$$U_1 = -BM_1^{-1} = \begin{pmatrix} 0 \\ I_m \\ 0 \\ Q_1 \end{pmatrix}, \quad U_2 = \tilde{B}^T = \begin{pmatrix} 0 \\ I_m \\ 0 \\ Q_2 \end{pmatrix}, \tag{21}$$

with the matrices $Q_1 = -M_2 R_{21} M_1^{-1}$ and $Q_2 = -R_{12}^T$. Moreover,

$$\mathbf{u}^* C\mathbf{u} = -(U_1^T\mathbf{u})^*(U_2^T\mathbf{u}) = -(\mathbf{u}_{\Gamma_1} + Q_1^T\mathbf{u}_{\Gamma_2})^*(\mathbf{u}_{\Gamma_1} + Q_2^T\mathbf{u}_{\Gamma_2}) = -\mathbf{y}_1^*\mathbf{y}_2,$$

with $\mathbf{y}_1 = \mathbf{u}_{\Gamma_1} + Q_1^T\mathbf{u}_{\Gamma_2}$ and $\mathbf{y}_2 = \mathbf{u}_{\Gamma_1} + Q_2^T\mathbf{u}_{\Gamma_2}$. Assuming the vector $\mathbf{u}$ has real components, $\mathbf{u}^T C\mathbf{u}$ would be positive only if the vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ would be pointing in two very different directions. This can only happen if the matrices Q_1 and Q_2 are very different in some sense. This situation seems highly unlikely provided the interpolation is accurate enough. Therefore, in the general case, we can expect the eigenvalues μ to have negative real part. In fact, an important result may be stated for the conforming case.

Lemma 5.1. *For a conforming mesh, the matrix C is symmetric negative semidefinite and the eigenvalues μ of interest are all strictly negative.*

Proof. In the conforming case, $M_1 = M_2$ and $R_{12} = R_{21} = I$ and one can easily verify that $\ker(\tilde{B}) = \ker(B^T)$ such that the subspaces $\mathcal{U}$ and $\mathcal{S}$ coincide. Moreover, C is obviously symmetric and $\mathbf{y}_1 = \mathbf{y}_2 = \mathbf{y} = \mathbf{u}_{\Gamma_1} - \mathbf{u}_{\Gamma_2}$. Thus, $\mathbf{u}^* C\mathbf{u} = -\|\mathbf{y}\|_2^2 < 0 \quad \forall \mathbf{u} \in \mathcal{U} = \mathcal{S}$. Consequently, $\mathcal{F} \subset (-\infty, 0)$ and in particular all generalized eigenvalues of interest are real strictly negative numbers. □

After investigating the sign of μ, we will now draw attention to its modulus.

5.4. Bounds on the generalized eigenvalues μ

In this subsection, we seek bounds on the eigenvalues μ of the generalized eigenvalue problem (20). We know that the matrix C has rank m, and we will assume the following ordering for its singular values

$$\sigma_1(C) \geq \sigma_2(C) \geq \cdots \geq \sigma_m(C) > \sigma_{m+1}(C) = \cdots \sigma_n(C) = 0.$$

Denoting $r = \operatorname{rank}(K) = n - s$, we will assume the following ordering for the eigenvalues of K

$$\lambda_1(K) \geq \lambda_2(K) \geq \cdots \geq \lambda_r(K) > \lambda_{r+1}(K) = \cdots = \lambda_n(K) = 0.$$

The next theorem provides an upper and lower bound on the modulus of μ.

Theorem 5.1. *Let μ be an eigenvalue of the generalized eigenvalue problem (20), then it satisfies the following bounds:*

$$\frac{\lambda_r(K)}{\sigma_1(C)} \leq |\mu| \leq \frac{\lambda_1(K)}{\sigma_m(C)}.$$

Proof. We define the following subspaces $\mathcal{V} = \mathbb{C}^n \setminus \ker(K)$ and $\mathcal{W} = \mathbb{C}^n \setminus \ker(\tilde{B})$ and let us note that $\mathcal{U} \subseteq \mathcal{V}$ and $\mathcal{U} \subseteq \mathcal{W}$. Without loss of generality, we assume all eigenvectors have unit norm. From the eigenvalue problem, we have

$$\|K\mathbf{u}\|_2 = |\mu| \|C\mathbf{u}\|_2.$$

We bound the left and right-hand sides as follows:

$$\lambda_1(K) = \max_{\substack{\mathbf{v}\in\mathbb{C}^n \\ \|\mathbf{v}\|_2=1}} \|K\mathbf{v}\|_2 \geq \max_{\substack{\mathbf{v}\in\mathcal{U} \\ \|\mathbf{v}\|_2=1}} \|K\mathbf{v}\|_2 \geq \|K\mathbf{u}\|_2 = |\mu| \|C\mathbf{u}\|_2,$$

$$\sigma_m(C)|\mu| = \min_{\substack{\mathbf{v}\in\mathcal{W} \\ \|\mathbf{v}\|_2=1}} |\mu| \|C\mathbf{v}\|_2 \leq \min_{\substack{\mathbf{v}\in\mathcal{U} \\ \|\mathbf{v}\|_2=1}} |\mu| \|C\mathbf{v}\|_2 \leq |\mu| \|C\mathbf{u}\|_2,$$

and we obtain the upper bound

$$|\mu| \leq \frac{\lambda_1(K)}{\sigma_m(C)}.$$

Similarly for the lower bound

$$\lambda_r(K) = \min_{\substack{\mathbf{v}\in\mathcal{V} \\ \|\mathbf{v}\|_2=1}} \|K\mathbf{v}\|_2 \leq \min_{\substack{\mathbf{v}\in\mathcal{U} \\ \|\mathbf{v}\|_2=1}} \|K\mathbf{v}\|_2 \leq \|K\mathbf{u}\|_2 = |\mu| \|C\mathbf{u}\|_2,$$

$$\sigma_1(C)|\mu| = \max_{\substack{\mathbf{v}\in\mathbb{C}^n \\ \|\mathbf{v}\|_2=1}} |\mu| \|C\mathbf{v}\|_2 \geq \max_{\substack{\mathbf{v}\in\mathcal{U} \\ \|\mathbf{v}\|_2=1}} |\mu| \|C\mathbf{v}\|_2 \geq |\mu| \|C\mathbf{u}\|_2,$$

and we obtain the lower bound

$$|\mu| \geq \frac{\lambda_r(K)}{\sigma_1(C)}. \quad \square$$

Theorem 5.1 provides an upper bound based on which we can choose $|\alpha|$. However, it is not a very practical result. Indeed, the computation of $\sigma_1(C)$ and $\sigma_m(C)$ may become expensive for very large problems. Therefore, we will instead seek bounds on those quantities. Let us first reorder the matrix C as a 2×2 block matrix such that C_{22}, the (2,2) block, is the only nonzero block. For non-conforming meshes at the interface, C_{22} is expressed as

$$C_{22} = \begin{pmatrix} -M_1 \\ M_2 R_{21} \end{pmatrix} M_1^{-1} \begin{pmatrix} I_m & -R_{12} \end{pmatrix} = -\underbrace{\begin{pmatrix} I_m \\ Q_1 \end{pmatrix}}_{Y_1} \underbrace{\begin{pmatrix} I_m & Q_2^T \end{pmatrix}}_{Y_2^T} \tag{22}$$

where $Q_1 = -M_2 R_{21} M_1^{-1}$ and $Q_2 = -R_{12}^T$. The matrices Y_1 and Y_2 are submatrices of the reordered factors U_1 and U_2, respectively, introduced in (21). Therefore, we already have a low-rank factorization for the matrix C_{22}. We already know that C_{22} has rank m and we would like to find an expression for its truncated singular value decomposition. For this purpose, we need orthonormal bases for Y_1 and Y_2. The next result, taken from [39], will be useful.

Lemma 5.2. *Let $Y \in \mathbb{C}^{n\times m}$ and $Q \in \mathbb{C}^{(n-m)\times m}$ with $n \geq m$ be given by*

$$Y = \begin{pmatrix} I_m \\ Q \end{pmatrix}.$$

Then the columns of $U = \begin{pmatrix} I_m \\ Q \end{pmatrix} (I_m + Q^*Q)^{-1/2}$ form an orthonormal basis for Y.

Proof. Clearly, the columns of U and Y span the same subspace. The small matrix $(I_m + Q^*Q)^{-1/2}$ is for the change of basis. Moreover, the columns of U are orthonormal. Indeed, by a direct computation

$$\begin{aligned} U^*U &= (I_m + Q^*Q)^{-1/2} \begin{pmatrix} I_m & Q^* \end{pmatrix} \begin{pmatrix} I_m \\ Q \end{pmatrix} (I_m + Q^*Q)^{-1/2} \\ &= (I_m + Q^*Q)^{-1/2} (I_m + Q^*Q)(I_m + Q^*Q)^{-1/2} = I_m. \quad \square \end{aligned}$$

A direct application of the previous lemma shows that the columns of the matrices

$$U = \begin{pmatrix} I_m \\ Q_1 \end{pmatrix} (I_m + Q_1^T Q_1)^{-1/2} = Y_1 (I_m + Q_1^T Q_1)^{-1/2} \text{ and}$$

$$V = \begin{pmatrix} I_m \\ Q_2 \end{pmatrix} (I_m + Q_2^T Q_2)^{-1/2} = Y_2 (I_m + Q_2^T Q_2)^{-1/2}$$

are orthonormal bases for Y_1 and Y_2, respectively. By using the latter formulas in equation (22), we obtain

$$C_{22} = -U (I_m + Q_1^T Q_1)^{1/2} (I_m + Q_2^T Q_2)^{1/2} V^T.$$

Let us now denote $Z_1 = (I_m + Q_1^T Q_1)^{1/2}$ and $Z_2 = (I_m + Q_2^T Q_2)^{1/2}$. These matrices are symmetric positive definite. We are only interested in the singular values of C_{22}, not in the left and right singular vectors. Clearly, the last expression shows that the singular values of C_{22} are the singular values of $Z_1 Z_2$. The next theorem provides bounds on those singular values.

Theorem 5.2. *Let $\lambda_1(M_2)$ and $\lambda_m(M_1)$ denote the largest eigenvalue of M_2 and smallest eigenvalue of M_1, respectively and let $\sigma(C)$ be a nonzero singular value of C. Then, it holds*

$$1 \leq \sigma(C) \leq \sqrt{\left(1 + \frac{\lambda_1^2(M_2)}{\lambda_m^2(M_1)} \|R_{21}\|_2^2\right)\left(1 + \|R_{12}\|_2^2\right)}.$$

Proof. To identify bounds on the singular values, we recall that the singular values of a matrix A are the square root of the eigenvalues of $A^T A$ or AA^T depending on the dimensions of the matrix. Then, notice that for all $\mathbf{v} \in \mathbb{R}^m$

$$\frac{\mathbf{v}^T Z_2 Z_1^2 Z_2 \mathbf{v}}{\|\mathbf{v}\|_2^2} = \frac{\mathbf{y}^T Z_1^2 \mathbf{y}}{\mathbf{y}^T Z_2^{-2} \mathbf{y}}$$

with $\mathbf{y} = Z_2 \mathbf{v}$. Recalling that Z_1 and Z_2 are symmetric matrices, the result follows immediately:

$$\sigma_m^2(Z_1)\sigma_m^2(Z_2) = \frac{\lambda_m(Z_1^2)}{\lambda_1(Z_2^{-2})} \leq \frac{\mathbf{y}^T Z_1^2 \mathbf{y}}{\mathbf{y}^T Z_2^{-2} \mathbf{y}} \leq \frac{\lambda_1(Z_1^2)}{\lambda_m(Z_2^{-2})} = \sigma_1^2(Z_1)\sigma_1^2(Z_2),$$

from which we conclude that if $\sigma(C)$ is a nonzero singular value of C_{22} and therefore of C, then

$$\sigma_m(Z_1)\sigma_m(Z_2) \leq \sigma(C) \leq \sigma_1(Z_1)\sigma_1(Z_2). \tag{23}$$

We do not necessarily have to compute these singular values. In fact, finding lower bounds on $\sigma_m(Z_1)$ and $\sigma_m(Z_2)$ and upper bounds on $\sigma_1(Z_1)$ and $\sigma_1(Z_2)$ is enough for our intended applications. Since $I_m + Q_1^T Q_1$ and $I_m + Q_2^T Q_2$ are symmetric positive definite, the matrix square root is well defined and the singular values (or eigenvalues) of Z_1 and Z_2 are expressed as

$$\sigma_i(Z_1) = \sqrt{1+\lambda_i(Q_1^TQ_1)} = \sqrt{1+\sigma_i^2(Q_1)} \quad i=1,\dots,m,$$
$$\sigma_i(Z_2) = \sqrt{1+\lambda_i(Q_2^TQ_2)} = \sqrt{1+\sigma_i^2(Q_2)} \quad i=1,\dots,m,$$

respectively. Then, we obtain straightforwardly

$$1 \le \sigma_i(Z_1) \le \sqrt{1+\|Q_1\|_2^2} \le \sqrt{1+\|M_1^{-1}\|_2^2\|M_2\|_2^2\|R_{21}\|_2^2}$$
$$= \sqrt{1+\frac{\lambda_1^2(M_2)}{\lambda_m^2(M_1)}\|R_{21}\|_2^2},$$
$$1 \le \sigma_i(Z_2) \le \sqrt{1+\|Q_2\|_2^2} = \sqrt{1+\|R_{12}\|_2^2},$$

from which we deduce lower and upper bounds on the smallest and largest singular values, respectively, of Z_1 and Z_2. The result of the theorem follows from the inequalities in (23). □

Combining Theorem 5.1 and Theorem 5.2 leads to the following useful result:

$$|\mu| \le \frac{\lambda_1(K)}{\sigma_m(C)} \le \lambda_1(K).$$

Although the analysis was not entirely trivial, the result is extremely simple. In addition, the upper bound may be very cheaply approximated using the Gershgorin circles. A much better estimate could be achieved with a few iterations of the Lanczos algorithm or even with the power method. See for instance [40] for a presentation of these methods. In practice, a tight upper bound is not required, which allows to use the cheapest available method. This result permits to choose α for instance as $\alpha = -u_K$ with u_K an upper bound on $\lambda_1(K)$. We can expect this choice to lead to a highly performing preconditioner while at the same time maintaining a moderate condition number for the $(1,1)$ block of the preconditioner, which will be useful when solving linear systems with the preconditioner.

Interestingly, the result indicates that the preconditioner should be unaffected by the conditioning of the INTERNODES matrix. In particular, the criterion does not explicitly depend on the matrices Q_1 and Q_2, which will be verified in the numerical experiments of Section 6, where an increased condition number by several orders of magnitude only leads to a very small increase of the number of iterations.

5.5. Solving linear systems with the preconditioning matrix

Now that we have a criterion for choosing α, it remains to solve efficiently linear systems with the preconditioning matrix defined in (14): that is $M\mathbf{x} = \mathbf{y}$, or equivalently

$$\begin{pmatrix} K+\alpha BM_1^{-1}\tilde{B} & 2B \\ 0 & -\alpha^{-1}M_1 \end{pmatrix}\begin{pmatrix}\mathbf{x}_1\\ \mathbf{x}_2\end{pmatrix} = \begin{pmatrix}\mathbf{y}_1\\ \mathbf{y}_2\end{pmatrix}.$$

Using the block-triangular structure, the second equation immediately yields $\mathbf{x}_2 = -\alpha M_1^{-1}\mathbf{y}_2$. Back-substituting in the first equation then leads to solving a linear system for $\mathbf{x}_1$

$$(K+\alpha BM_1^{-1}\tilde{B})\mathbf{x}_1 = \mathbf{y}_1 + 2\alpha BM_1^{-1}\mathbf{y}_2 = \tilde{\mathbf{y}}_1.$$

Solving efficiently this linear system is the main difficulty of augmented Lagrangian preconditioners. Our efforts are hampered for three reasons. Firstly, the potential singularity of K prevents us from using the Woodbury matrix identity as a starting point for designing inexact solves. Secondly, the low rank term $BM_1^{-1}\tilde{B}$ changes during the course of the contact algorithm. Therefore, preconditioned iterative schemes would require recomputing a preconditioner between the iterations of the contact algorithm, which may be too expensive. The lack of symmetry then further adds to the computational cost of the iterative scheme. Thirdly, the matrix cannot be formed explicitly for very large contact problems. Indeed, some of the blocks of the matrices B and $\tilde{B}$ contain interpolation matrices and their definition involves Φ_{MM}^{-1}, which is completely dense. Moreover, any sparsity left in the nonzero blocks of B and $\tilde{B}$ is destroyed when carrying out the multiplication with M_1^{-1}. Designing a strategy addressing all three difficulties together is increasingly challenging. Solution methods for related problems can be found in [26,27,41] and are essentially based on multigrid methods. The approach we propose in this paper is well-suited for medium size applications. Its extension to larger applications will be discussed subsequently. Our strategy consists in two steps. Firstly, the stiffness matrix K is replaced by $\tilde{K} = K + \epsilon I_n$ with $\epsilon > 0$. Adding the term ϵI_n to K leads to a symmetric positive definite (and therefore invertible) matrix. Hence, $\tilde{K}$ admits a Cholesky factorization $\tilde{K} = LL^T$ where L is a lower triangular matrix. The Cholesky factorization is unique for symmetric positive definite matrices. Our strategy then relies on exploiting the sparsity of the blocks B and $\tilde{B}$. At the heart of our method lies the following theorem.

Theorem 5.3. *Let the matrix $\tilde{K} + \alpha BM_1^{-1}\tilde{B}$ be ordered such that*

$$\tilde{K} + \alpha BM_1^{-1}\tilde{B} = \begin{pmatrix}\tilde{K}_{11} & \tilde{K}_{12}\\ \tilde{K}_{21} & \tilde{K}_{22}\end{pmatrix} - \alpha\begin{pmatrix}0 & 0\\ 0 & Y_1Y_2^T\end{pmatrix}$$

with Y_1 and Y_2 defined in (22). Moreover, let

$$L = \begin{pmatrix}L_{11} & 0\\ L_{21} & L_{22}\end{pmatrix}$$

be the Cholesky factor of $\tilde{K}$ and let us denote $G = I - \alpha L_{22}^{-1}Y_1Y_2^TL_{22}^{-T}$. Then, $\tilde{K} + \alpha BM_1^{-1}\tilde{B}$ admits a block LDL^T factorization given by

$$\tilde{K} + \alpha BM_1^{-1}\tilde{B} = \begin{pmatrix}L_{11} & 0\\ L_{21} & L_{22}\end{pmatrix}\begin{pmatrix}I & 0\\ 0 & G\end{pmatrix}\begin{pmatrix}L_{11}^T & L_{21}^T\\ 0 & L_{22}^T\end{pmatrix} = LDL^T.$$

Proof. Since $\tilde{K} = LL^T$, a simple substitution leads to

$$\tilde{K} + \alpha BM_1^{-1}\tilde{B} = LL^T + \alpha BM_1^{-1}\tilde{B} = L(I + \alpha L^{-1}BM_1^{-1}\tilde{B}L^{-T})L^T.$$

The sparsity of the matrices B and $\tilde{B}$ can be used to simplify the computation of $L^{-1}BM_1^{-1}\tilde{B}L^{-T}$:

$$L^{-1}BM_1^{-1}\tilde{B}L^{-T}$$
$$= \begin{pmatrix}L_{11}^{-1} & 0\\ -L_{22}^{-1}L_{21}L_{11}^{-1} & L_{22}^{-1}\end{pmatrix}\begin{pmatrix}0 & 0\\ 0 & -Y_1Y_2^T\end{pmatrix}\begin{pmatrix}L_{11}^{-T} & -L_{11}^{-T}L_{21}^TL_{22}^{-T}\\ 0 & L_{22}^{-T}\end{pmatrix}$$
$$= \begin{pmatrix}0 & 0\\ 0 & -L_{22}^{-1}Y_1Y_2^TL_{22}^{-T}\end{pmatrix}.$$

Therefore, we obtain

$$\tilde{K} + \alpha BM_1^{-1}\tilde{B} = \begin{pmatrix}L_{11} & 0\\ L_{21} & L_{22}\end{pmatrix}\begin{pmatrix}I & 0\\ 0 & I-\alpha L_{22}^{-1}Y_1Y_2^TL_{22}^{-T}\end{pmatrix}\begin{pmatrix}L_{11}^T & L_{21}^T\\ 0 & L_{22}^T\end{pmatrix}.$$

□

Consequently, assuming $G = I - \alpha L_{22}^{-1}Y_1Y_2^TL_{22}^{-T}$ is invertible, the inverse of $\tilde{K} + \alpha BM_1^{-1}\tilde{B}$ admits the explicit expression

$$(\tilde{K} + \alpha BM_1^{-1}\tilde{B})^{-1}$$
$$= \begin{pmatrix}L_{11}^{-T} & -L_{11}^{-T}L_{21}^TL_{22}^{-T}\\ 0 & L_{22}^{-T}\end{pmatrix}\begin{pmatrix}I & 0\\ 0 & G^{-1}\end{pmatrix}\begin{pmatrix}L_{11}^{-1} & 0\\ -L_{22}^{-1}L_{21}L_{11}^{-1} & L_{22}^{-1}\end{pmatrix}$$
$$= L^{-T}D^{-1}L^{-1}.$$

We will denote b_1 and b_2 the sizes of L_{11} and L_{22}, respectively with $b_2 \ll b_1$. Consequently, solving a linear system with $\tilde{K} + \alpha BM_1^{-1}\tilde{B}$ only requires the solution of two large triangular systems and another very small linear system with the matrix G. In linear elasticity without any remeshing, the stiffness matrix K does not change during the iterations of the contact algorithm. Thus, we may conveniently compute only a single Cholesky factorization of $\tilde{K}$ and reuse it during all subsequent iterations of the contact algorithm for solves with the preconditioning

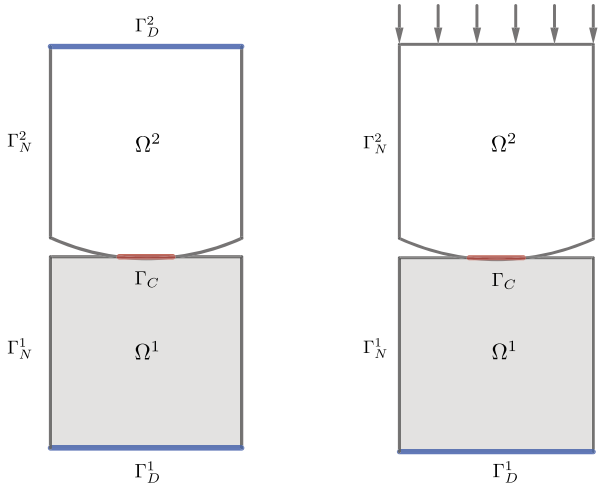

Fig. 3. Hertzian contact problems, at left the first configuration, at right the second one.

matrix. On the other hand, the matrix G is completely dense. However, if the problem is small enough, it can be computed explicitly and direct methods can be used to solve these small linear systems. It must be noted that the Cholesky factorization is not the only factorization possible. The proof can be easily adjusted to accommodate LU or LDL^T factorizations of $\tilde{K}$. However, the Cholesky factorization is the natural choice for obvious storage reasons.

For very large problems, resorting to a Cholesky factorization or even forming the matrix G becomes infeasible. However, the same strategy could potentially be used with only a few adjustments. First of all, an incomplete Cholesky factorization of $\tilde{K}$ can be used instead of the complete one. Secondly, linear systems with the small matrix G may be solved iteratively. However, using an incomplete Cholesky factorization of $\tilde{K}$ will surely have a negative impact on the clustering of the eigenvalues of the preconditioned matrix. Some early numerical experiments are reported in the next section for a quantitative assessment.

6. Numerical experiments

In this section, the quality of the preconditioner is tested on contact problems of increasingly large size. We will consider as benchmark the classic Hertzian contact problem between two elastic bodies with two different loading conditions.[1] The considered geometry and boundary conditions are shown in Fig. 3.

The first body is represented in gray and the second one in white. Between them lies the potential contact interface Γ_C represented in red. The first body is subjected to homogeneous Dirichlet boundary conditions on its base (in blue in the figure). The only difference between our two configurations lies in the boundary conditions of the upper body. In the first configuration (Fig. 3, left), we prescribe non-homogeneous Dirichlet boundary conditions on its top edge. In the second configuration (Fig. 3, right), a uniform traction is instead applied on this same edge. Therefore, the second body is only subjected to Neumann boundary conditions. This will result in a singularity of the stiffness matrix K. The geometry has been discretized with $\mathbb{P}_1$ finite elements of mesh size h. To solve the contact problem we adopt an algorithm, here referred to as the *contact algorithm*, which usually requires solving a sequence of linear systems and not just a single one. This algorithm calls many different functions that include:

1. Computing the radii of radial basis functions.
2. Assembling the interpolation matrices, interface mass matrices and stiffness matrix.
3. Solving the linear systems with the INTERNODES matrix.
4. Verifying the convergence of the algorithm.

[1] The code used in this work is freely available at the following address: https://c4science.ch/diffusion/CONTACTINTERNODES/.

Table 2
Mesh sizes and block sizes for configuration 1.

h	n	m
0.1	530	18
0.05	1 956	34
0.01	46 484	114
0.005	184 134	88

Table 3
Mesh sizes and block sizes for configuration 2.

h	n	m
0.1	552	18
0.05	1 998	34
0.01	46 686	114
0.005	184 536	88

The maximum number of iterations for the contact algorithm was set to 10. Most of the time, it converged within two to three iterations with the exception of the run with the very small mesh size, here $h = 0.005$. The mesh sizes and the corresponding block sizes of the matrix A we have considered are reported in Tables 2 and 3.

The block size n depends on the total number of unknowns, thus, the second configuration leads to a slightly larger n due to the Neumann boundary conditions. The block size m is equal to the number of degrees of freedom on Γ_C^1 and changes at each iteration of the contact algorithm. The values reported in Tables 2 and 3 correspond to the first iteration of the contact algorithm. The block size is controlled by radial basis function interpolation requirements and the interpolation matrices were constructed following the strategies presented in Section 2. We will solve the contact problem for these two configurations using a right preconditioned GMRES method [42] for solving the linear systems. Figs. 4 and 5 report the decrease of the residual norm for configurations 1 and 2, respectively, corresponding to the first iteration of the contact algorithm. In all numerical experiments, an absolute stopping criterion was used on the norm of the residual vector with a tolerance of 10^{-8} and a GMRES restart value of 100. The elastic modulus was $E = 30 \times 10^9$ Pa and the Poisson ratio was $\nu = 0.2$. As for the value of α, it is computed internally with the matrix infinity-norm used to provide a cheap upper bound on the largest eigenvalue of the stiffness matrix ($\alpha = -\|K\|_\infty$). Since K is symmetric, the matrix 1-norm yields the same result. For 2D problems, the largest eigenvalue of the stiffness matrix is independent of the mesh size [43]. Our criterion for computing α captures this feature, leading to a roughly constant numerical value $\alpha = -7.5023$.

Figs. 4 and 5 firstly indicate that the number of iterations needed to reach the prescribed tolerance is almost independent of the mesh size. This is an extremely precious property when using an iterative solver such as GMRES. As its cost increases at each iteration, the method is efficient only if the iteration count remains small. Secondly, the preconditioner is robust: cases with singular or invertible stiffness matrices are handled equally well, which does not come as a surprise as our preconditioner was designed to handle singular stiffness matrices. Thirdly, we noticed that the quality of the preconditioner was not affected by the geometry. More complicated case studies with several points of contact were also solved within a few iterations.

When the size of the matrix $\tilde{K}$ is very large (especially in 3D problems) the Cholesky factorization could become prohibitively expensive. In this case an incomplete Cholesky factorization could be used instead of the complete one. Although the experiments should be carried out on very large matrices, we report here some preliminary results on small

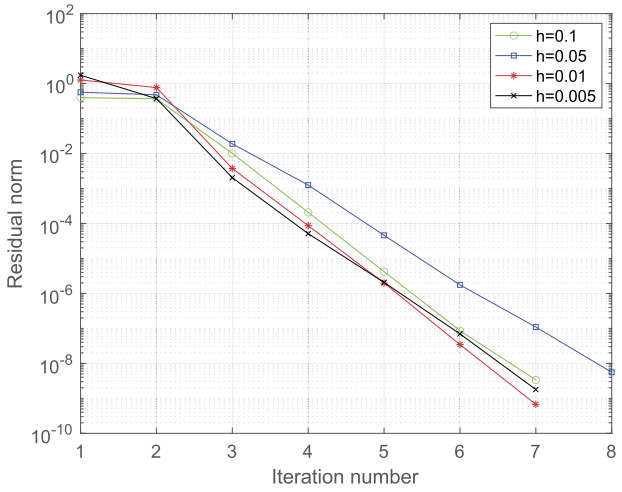

Fig. 4. Residual norm for configuration 1 using a complete Cholesky factorization.

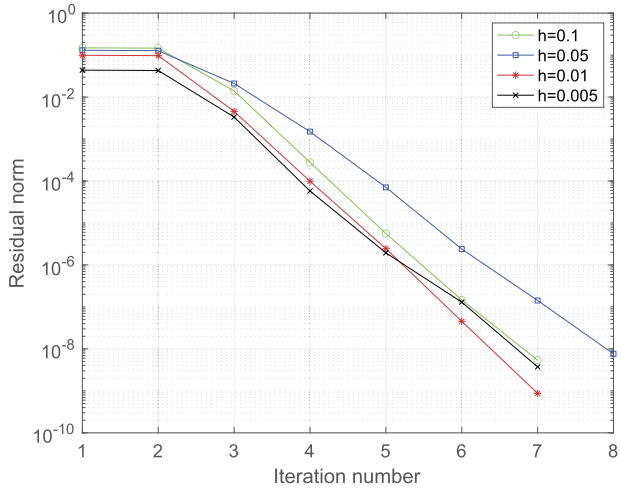

Fig. 5. Residual norm for configuration 2 using a complete Cholesky factorization.

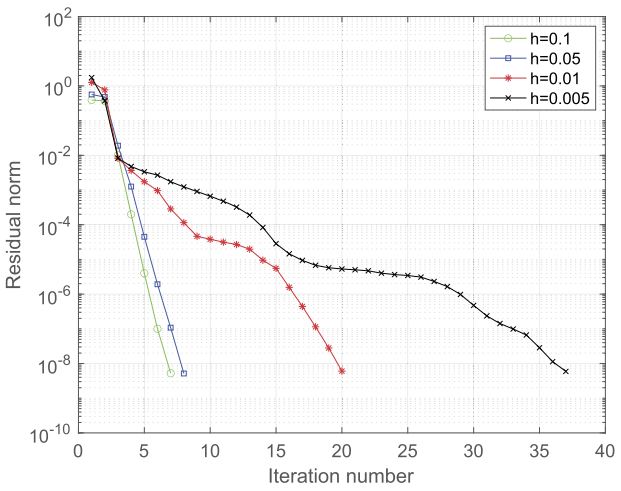

Fig. 6. Residual norm for configuration 1 using an incomplete Cholesky factorization.

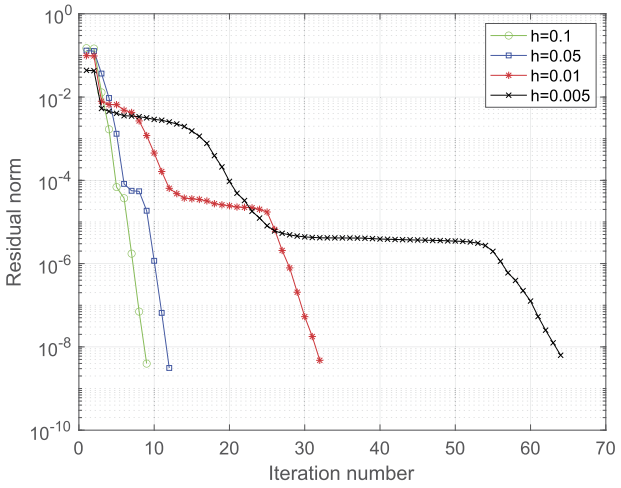

Fig. 7. Residual norm for configuration 2 using an incomplete Cholesky factorization.

matrices. We emphasize that these are early numerical results and are not meant to draw definite conclusions. We did not carry out an exhaustive grid search of parameters of the incomplete Cholesky factorization. Instead, a single dropping tolerance of 10^{-4} is used to compute the incomplete factorization. The results for both configurations are shown in Figs. 6 and 7 and the iteration counts are listed in Table 4. For the first configuration, using an incomplete factorization still leads to a very good preconditioner, however, the iteration count increases with the size of the matrix. For the configuration having a singular stiffness matrix, more serious issues are encountered: the incomplete factorization produces nonpositive (most likely zero) pivots and consequently the factorization fails. This is expected since the perturbed stiffness matrix, defined as $\tilde{K} = K + \epsilon I_n$ with $\epsilon = 10^{-8}$, will see the perturbation ϵI_n wiped out with a dropping tolerance of 10^{-4}. Without it, the positive definiteness of $\tilde{K}$ is lost, the matrix becomes singular and the factorization fails. An easy workaround is to increase the magnitude of the perturbation by choosing ϵ larger than the drop tolerance. For instance, setting $\epsilon = 10^{-3}$ instead of 10^{-8}. While it successfully removes the issue, it also further degrades the quality of the preconditioner, as testified by the increased number of iterations needed to satisfy the stopping criterion for a given tolerance. Moreover, we notice that although the Cholesky factorization exists for every symmetric positive definite matrix, the incomplete factorization may fail. More investigation is needed before drawing definite conclusions on the prospects of incomplete factorizations. We certainly do not claim incomplete factorizations are the best strategy. On the contrary, future work should explore possible usage of algebraic multigrid methods for inner solves with the matrix $\tilde{K} + \alpha B M_1^{-1} \tilde{B}$.

Table 4
Iteration counts for configurations 1 and 2 using an incomplete Cholesky factorization.

h	Configuration 1	Configuration 2
0.1	7	9
0.05	8	12
0.01	20	32
0.005	37	64

According to Wathen [44], the cost of solving a linear system with the preconditioning matrix should balance the cost of computing a matrix-vector multiplication with the coefficient matrix. These two cost components are briefly analyzed here. Since $m \ll n$ and due to the sparsity of the matrices B and $\tilde{B}$, matrix-vector multiplications with the INTERNODES matrix A are dominated by the sparse matrix-vector multiplications with the stiffness matrix K. Thus, computing $A\mathbf{x}$ is expected to cost $O(\mathrm{nzz}(K))$ where nzz denotes the number of nonzero entries. The cost for solving a linear system with the preconditioning matrix M is dominated by the inner solve with the matrix $K + \alpha B M_1^{-1} \tilde{B}$. Our

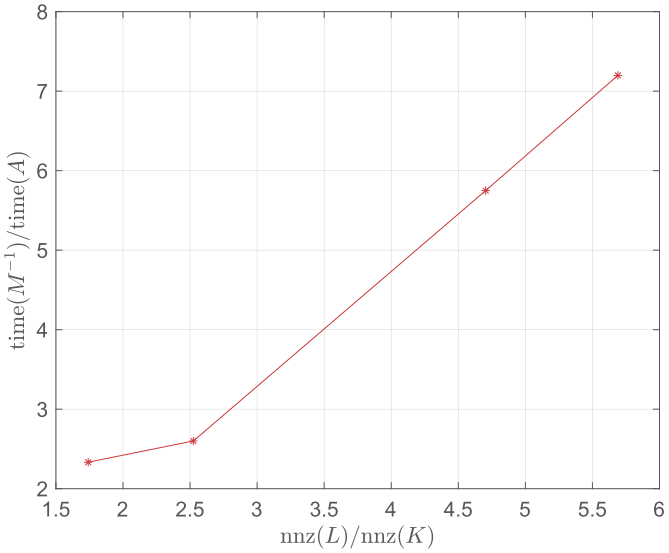

Fig. 8. Ratio of measured computational times for an application of M^{-1} and A as a function of the ratio of number of nonzero entries in L and K.

Table 5
Spectral condition number of Q_1, Q_2 and A.

h	$\kappa(Q_1)$	$\kappa(Q_2)$	$\kappa(A)$
0.1	1.12×10^{6}	4.86×10^{5}	8.31×10^{3}
0.05	3.37×10^{7}	1.89×10^{7}	2.78×10^{5}
0.01	2.22×10^{13}	1.66×10^{13}	5.08×10^{6}
0.005	7.37×10^{14}	5.49×10^{14}	3.89×10^{7}

strategy for medium size problems relies on solving two triangular systems with the Cholesky factor L of a perturbed stiffness matrix and one much smaller linear system. Thus, the expected cost is $O(\text{nzz}(L))$. Despite sparse reordering strategies, $\text{nzz}(L) > \text{nzz}(K)$, such that applying M^{-1} is more expensive than computing matrix-vector products. Fig. 8 represents the ratio of computational times measured for an application of M^{-1} and A as a function of the ratio of the number of nonzero entries in L and K. This experiment reuses the same mesh sizes. In order to measure the application cost of M^{-1} independently of the number of iterations of the contact algorithm, the preconditioner setup time is not included. Then, as expected, the trend is linear if the problem is large enough.

Although incomplete factorizations reduce the preconditioner's footprint, they also increase the iteration count and the memory consumption of GMRES. For all our experiments on medium size problems, the higher application cost of M^{-1} using complete factorizations always paid off and largely counter-balanced the increased number of iterations induced by incomplete ones.

The theoretical results of Section 5.4 indicated that our criterion for choosing α did not explicitly depend on the matrices Q_1 and Q_2. We propose to verify it experimentally by artificially increasing the condition number of the matrices. This test case illustrates the theoretical findings beyond contact problems. We recall that the matrices Q_1 and Q_2 are defined as $Q_1 = -M_2 R_{21} M_1^{-1}$ and $Q_2 = -R_{12}^T$. The matrices R_{21} and R_{12} are replaced by increasingly ill-conditioned matrices with prescribed singular values. The spectral condition number of these matrices and of the saddle point matrix itself is reported in Table 5 for the different mesh sizes.

Our criterion for choosing α only depends on the stiffness matrix and is roughly constant for 2D applications. For the sake of the comparison, it was fixed at $\alpha = -10$ independently of the mesh size. The right-hand side vector was taken as the vector of all ones and the parameters of GMRES were kept unchanged (tolerance of 10^{-8} and restart of 100). The decrease of the residual norm is shown in Fig. 9. Despite the increased

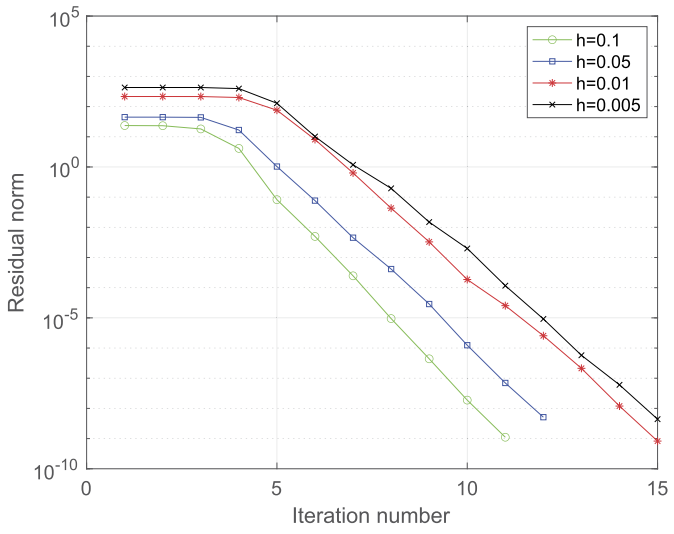

Fig. 9. Residual norm for increasingly ill-conditioned matrices Q_1 and Q_2.

Table 6
Iteration count for the preconditioners (19) and (24).

Preconditioner	None	$\mathcal{P}_{Aug-}$	$\mathcal{P}_{Aug+}$	$\mathcal{D}_{Aug}$	M_{α_+}	M_{α_-}
Iterations	638	56	39	29	28	8

condition number by several orders of magnitude, the iteration count increases only very mildly, supporting the theoretical findings and the robustness of our criterion.

In the next experiment, we show how our tuning strategy greatly improves the quality of the preconditioners proposed in [36] and [38]. Our testing includes two original block-triangular preconditioners developed in [36] (called $\mathcal{P}_{Aug-}$ and $\mathcal{P}_{Aug+}$), one block-diagonal preconditioner (called $\mathcal{D}_{Aug}$) proposed by the same author and one block-triangular preconditioner developed in [38] ($\mathcal{A}$ denoted here M_α) with W replaced by $\alpha^{-1}W$ in order to highlight the importance of the weighting factor α. As usual, we set $W = M_1$. They are:

$$\mathcal{P}_{Aug-} = \begin{pmatrix} K + BW^{-1}\tilde{B} & B \\ 0 & -W \end{pmatrix}, \quad \mathcal{P}_{Aug+} = \begin{pmatrix} K + BW^{-1}\tilde{B} & B \\ 0 & W \end{pmatrix},$$
$$\mathcal{D}_{Aug} = \begin{pmatrix} K + BW^{-1}\tilde{B} & 0 \\ 0 & W \end{pmatrix}, \quad M_\alpha = \begin{pmatrix} K + \alpha BW^{-1}\tilde{B} & 2B \\ 0 & -\alpha^{-1}W \end{pmatrix}. \tag{24}$$

Note that the preconditioners $\mathcal{P}_{Aug-}$, $\mathcal{P}_{Aug+}$ and M_α have the same sparsity pattern and differ only in a few constants multiplying some of the blocks. For the preconditioner M_α, the default parameter choice $\alpha_+ = 1$ is compared to our scaling strategy with $\alpha_- = -\|K\|_\infty$. Fig. 10 reports the results for an intermediate mesh size $h = 0.05$. The material parameters are as in the previous tests. The right preconditioned GMRES method was again used with a restart of 100 and a tolerance of 10^{-8}. The results illustrate the effectiveness of our strategy and reveal that the parameters in the preconditioner have a significant impact. Namely, for Cao's native preconditioners and Liu's preconditioner M_α with $\alpha_+ = 1$, convergence is initially extremely slow before reaching a superconvergence regime. Thanks to our specific scaling, this regime is reached much faster. The iteration counts for the different preconditioning strategies are listed in Table 6. The iteration count for the unpreconditioned GMRES method is noted for comparison.

When solving the linear system (13), a direct method could be used (at least for small to medium size problems) instead of our preconditioned GMRES method. Thus, we are interested in comparing the performance of these two methods in the context of the contact problem. In MATLAB, a direct method is called when using the backslash command. Due to the properties and sparsity of the INTERNODES matrix,

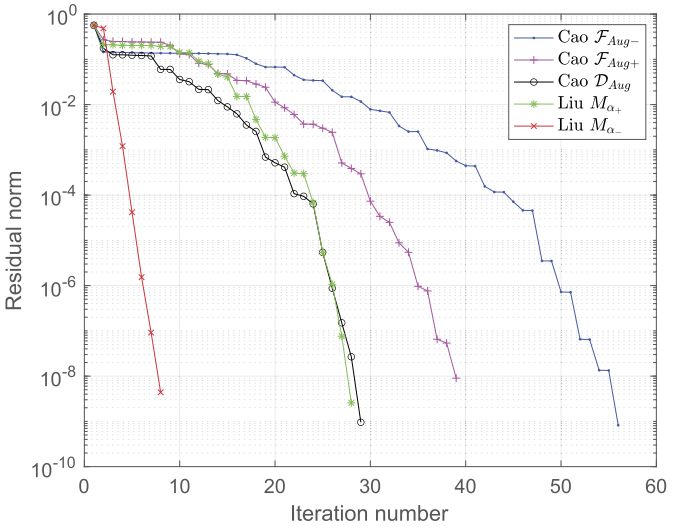

Fig. 10. Performance comparison for the preconditioners (19) and (24).

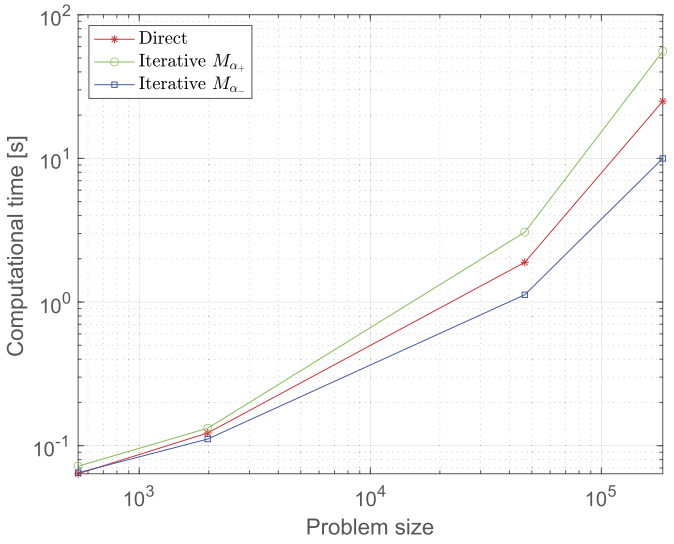

Fig. 11. Computational times for the contact algorithm.

Table 7
Computational times and speedup factors.

h	0.1	0.05	0.01	0.005
Time direct [s]	0.064	0.123	1.89	24.99
Time iterative M_{α_+} [s]	0.072	0.132	3.06	55.61
Time iterative M_{α_-} [s]	0.065	0.111	1.12	9.98

the backslash command calls some state-of-the-art implementation of a multifrontal method [45]. On the contrary, our method uses an in-house implementation of preconditioned GMRES. We did not use the built-in *gmres* function of MATLAB because we lacked control over the stopping criterion. Moreover, the MATLAB function uses the left preconditioner whereas we are using the right one. Our implementation is very basic and its performance could certainly be improved. All the experiments are carried out in MATLAB R2021a on MacOS with a Dual-Core Intel Core i7 processor with 2.2 GHz of processing speed and 8 GB of RAM. In Fig. 11, the computational times for the entire contact algorithm, including the preconditioner setup time for the iterative solution, are reported for configuration 1. To further highlight the impact of the scaling parameter, the preconditioner with our choice of scaling ($\alpha_- = -\|K\|_\infty$) is compared with the default choice ($\alpha_+ = 1$). These preconditioners have exactly the same block structure and consequently any difference in performance stems from the different number of iterations. Some important observations must be made. First of all, the computational times are always relatively small for both implementations (direct and iterative) and it is extremely challenging to outperform direct methods for applications where they excel: moderate size finite element matrices for 2D problems. For example, for the mesh size $h = 0.01$ associated with a matrix of size over 46 000, three large linear systems are solved and the contact algorithm computes the solution in roughly 1.89 s with a direct method. Despite their remarkable performance, our preconditioned GMRES implementation with a suitably scaled preconditioner always outperforms direct methods by a large margin on small mesh sizes. The computational times are also reported in Table 7. Since the number of linear systems solved in the sequence is different for different mesh sizes, one should not attempt to fit a trendline over the values reported.

The speedup is expected to increase further with the size of the matrix and the number of linear systems solved in the sequence. Since our method uses a single Cholesky factorization for $\tilde{K}$, its cost is amortized over several linear systems. In comparison, direct methods will recompute a factorization at each iteration. Finally, qualitatively speaking, the numerical solutions obtained with both methods were indistinguishable. The euclidean norm of the error was at most of the order 10^{-7}, suggesting that our tolerance on the residual was adequate.

7. Conclusions and new perspectives

In this work, a highly efficient preconditioner was designed for solving the sequence of linear systems arising from the application of the INTERNODES method to linear elastic problems in contact mechanics. The saddle point type structure inherited from the variational problem and the properties of the stiffness matrix naturally led to considering an augmented Lagrangian preconditioner. We have discussed how the preconditioner could be suitably tuned and we have analyzed the spectrum of the preconditioned matrix. Taking advantage of the sparsity structure of the matrices, we then proposed an algorithm to efficiently solve linear systems with the preconditioning matrix. Assuming a linear elastic constitutive model, our strategy relies on a unique Cholesky factorization for the stiffness matrix $\tilde{K}$ which is computed once and for all. Resorting to a complete factorization allows the implementation of a nearly ideal preconditioner. Numerical experiments confirmed the quality and robustness of the preconditioner. In addition, they revealed that the convergence rate was independent of the mesh size and our preconditioned iterative scheme outperformed state-of-the-art sparse direct solvers.

To avoid excessive memory usage, we have also discussed possible extensions to incomplete factorizations. We have shown experimentally that incomplete factorizations could still lead to an efficient preconditioner. However, pivot breakdowns were also occasionally experienced during the factorization process. Such breakdowns are well known for applications in structural mechanics and robust incomplete factorizations could provide a possible remedy [46]. Yet, another unfortunate feature common to all incomplete factorizations is the increase of the iteration count with the problem size. Other solution techniques were already examined in an extension of our work [15]. Some of these Cholesky-free methods seem very promising for large 3D problems but there is still much empirical work left in determining the best strategy. Cholesky-free methods would also certainly be favored in case the stiffness matrix must be reassembled at each iteration of the contact algorithm, for example in case of nonlinear constitutive models.

Data availability

The code and data are freely available at Code Ocean.

Acknowledgements

We are grateful to the two anonymous referees for their careful reading and helpful suggestions. Simone Deparis has been supported by the Swiss National Science Foundation under project FNS 200021197021.

Appendix A. Proof of Theorem 4.1

Let

$$B^T = U_B[\Sigma_B\ 0]V_B^T \qquad \text{and} \qquad \tilde{B} = U_{\tilde{B}}[\Sigma_{\tilde{B}}\ 0]V_{\tilde{B}}^T$$

be the singular value decomposition of B^T and $\tilde{B}$ respectively. $U_B, U_{\tilde{B}} \in \mathbb{R}^{m\times m}$ and $V_B, V_{\tilde{B}} \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $\Sigma_B, \Sigma_{\tilde{B}} \in \mathbb{R}^{m\times m}$ are diagonal matrices containing the singular values. The matrices V_B and $V_{\tilde{B}}$ are further partitioned as

$$V_B = [V_{B_1} V_{B_2}] \text{ and } V_{\tilde{B}} = [V_{\tilde{B}_1} V_{\tilde{B}_2}]$$

with V_{B_2} and $V_{\tilde{B}_2}$ two bases for the kernels of B^T and $\tilde{B}$, respectively. We now form the orthogonal matrices S_1 and S_2 such that

$$S_1 = \begin{pmatrix} V_B^T & 0 \\ 0 & U_{\tilde{B}}^T \end{pmatrix} \quad S_2 = \begin{pmatrix} V_{\tilde{B}} & 0 \\ 0 & U_B \end{pmatrix}$$

and consider the matrix $\tilde{A} = S_1 A S_2$ explicitly given by

$$\begin{aligned}\tilde{A} = S_1 A S_2 &= \begin{pmatrix} V_B^T & 0 \\ 0 & U_{\tilde{B}}^T \end{pmatrix}\begin{pmatrix} K & B \\ \tilde{B} & 0 \end{pmatrix}\begin{pmatrix} V_{\tilde{B}} & 0 \\ 0 & U_B \end{pmatrix} \\ &= \begin{pmatrix} V_B^T K V_{\tilde{B}} & V_B^T B U_B \\ U_{\tilde{B}}^T \tilde{B} V_{\tilde{B}} & 0 \end{pmatrix} \\ &= \begin{pmatrix} V_{B_1}^T K V_{\tilde{B}_1} & V_{B_1}^T K V_{\tilde{B}_2} & \Sigma_B \\ V_{B_2}^T K V_{\tilde{B}_1} & V_{B_2}^T K V_{\tilde{B}_2} & 0 \\ \Sigma_{\tilde{B}} & 0 & 0 \end{pmatrix}.\end{aligned}$$

Since S_1 and S_2 are orthogonal matrices, A and $\tilde{A}$ have the same singular values and in particular $\kappa(\tilde{A}) = \kappa(A)$. The structure of $\tilde{A}$ will be used to find a lower bound on the spectral condition number of A. We first prove that $\kappa(A) \geq \kappa_{\tilde{B}}(K)\max\{1, \frac{\|B\|_2}{\sigma_{1,\tilde{B}}}, \frac{\|\tilde{B}\|_2}{\sigma_{1,\tilde{B}}}\}$. All proofs are based on the following facts:

$$\sigma_1(A) = \sigma_1(\tilde{A}) = \max_{\|\mathbf{x}\|_2=1} \|\tilde{A}\mathbf{x}\|_2 \geq \|\tilde{A}\mathbf{y}\|_2 \text{ for some } \mathbf{y} \text{ such that } \|\mathbf{y}\|_2 = 1,$$

$$\sigma_{n+m}(A) = \sigma_{n+m}(\tilde{A}) = \min_{\|\mathbf{x}\|_2=1} \|\tilde{A}\mathbf{x}\|_2 \leq \|\tilde{A}\mathbf{y}\|_2 \text{ for some } \mathbf{y} \text{ such that } \|\mathbf{y}\|_2 = 1.$$

The strategy consists in choosing the vectors $\mathbf{y}$ cleverly such that the resulting lower bound on the condition number is tight.

Lower bound on $\sigma_1(A)$:

Since the contributions of K, B and $\tilde{B}$ must be captured, let us consider different choices.

1. Choose $\mathbf{y}^T = (\mathbf{y}_1^T, \mathbf{0}^T, \mathbf{0}^T)$ with $\|\mathbf{y}_1\|_2 = 1$ such that $\|\Sigma_{\tilde{B}}\mathbf{y}_1\|_2 = \|\tilde{B}\|_2$. If the singular values of $\tilde{B}$ have been placed in decreasing order along the diagonal of $\Sigma_{\tilde{B}}$, then $\mathbf{y}_1 = \mathbf{e}_1$, the first vector of the canonical basis of $\mathbb{R}^m$. Thus,

$$\begin{aligned}\|\tilde{A}\mathbf{y}\|_2^2 &= \|V_B^T K V_{\tilde{B}_1}\mathbf{y}_1\|_2^2 + \|\Sigma_{\tilde{B}}\mathbf{y}_1\|_2^2 \\ &= \|KV_{\tilde{B}_1}\mathbf{y}_1\|_2^2 + \|\tilde{B}\|_2^2 \\ &\geq \|\tilde{B}\|_2^2.\end{aligned}$$

2. Choose $\mathbf{y}^T = (\mathbf{0}^T, \mathbf{y}_2^T, \mathbf{0}^T)$ with $\|\mathbf{y}_2\|_2 = 1$ such that

$$\|KV_{\tilde{B}_2}\mathbf{y}_2\|_2 = \min_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(\tilde{B})}} \|K\mathbf{x}\|_2 = \sigma_{1,\tilde{B}}.$$

Hence, $\|\tilde{A}\mathbf{y}\|_2^2 = \|V_B^T K V_{\tilde{B}_2}\mathbf{y}_2\|_2^2 = \|KV_{\tilde{B}_2}\mathbf{y}_2\|_2^2 = \sigma_{1,\tilde{B}}^2$.

3. Choose $\mathbf{y}^T = (\mathbf{0}^T, \mathbf{0}^T, \mathbf{y}_3^T)$ with $\|\mathbf{y}_3\|_2 = 1$ such that $\|\Sigma_B\mathbf{y}_3\|_2 = \|B\|_2$. If the singular values of B have been placed in decreasing order along the diagonal of Σ_B, then $\mathbf{y}_3 = \mathbf{e}_1$, the first vector of the canonical basis of $\mathbb{R}^m$. Then, we obtain straightforwardly $\|\tilde{A}\mathbf{y}\|_2^2 = \|\Sigma_B\mathbf{y}_3\|_2^2 = \|B\|_2^2$.

Since all particular choices yield a lower bound, we deduce that $\sigma_1(A) \geq \max\{\sigma_{1,\tilde{B}}, \|B\|_2, \|\tilde{B}\|_2\}$.

Upper bound on $\sigma_{n+m}(A)$:

This time we simply take $\mathbf{y}^T = (\mathbf{0}^T, \mathbf{y}_2^T, \mathbf{0}^T)$ with $\|\mathbf{y}_2\|_2 = 1$ such that

$$\|KV_{\tilde{B}_2}\mathbf{y}_2\|_2 = \min_{\substack{\|\mathbf{x}\|_2=1 \\ \mathbf{x}\in\ker(\tilde{B})}} \|K\mathbf{x}\|_2 = \sigma_{n,\tilde{B}}.$$

Hence, $\|\tilde{A}\mathbf{y}\|_2^2 = \|V_B^T K V_{\tilde{B}_2}\mathbf{y}_2\|_2^2 = \|KV_{\tilde{B}_2}\mathbf{y}_2\|_2^2 = \sigma_{n,\tilde{B}}^2$. Consequently, $\sigma_{n+m}(A) \leq \sigma_{n,\tilde{B}}$.

Gathering all the results, we finally obtain

$$\begin{aligned}\kappa(A) = \kappa(\tilde{A}) = \frac{\sigma_1(A)}{\sigma_{n+m}(A)} &\geq \frac{\max\{\sigma_{1,\tilde{B}}, \|B\|_2, \|\tilde{B}\|_2\}}{\sigma_{n,\tilde{B}}} \\ &= \kappa_{\tilde{B}}(K)\max\{1, \frac{\|B\|_2}{\sigma_{1,\tilde{B}}}, \frac{\|\tilde{B}\|_2}{\sigma_{1,\tilde{B}}}\}\end{aligned}$$

Since the singular values of $\tilde{A}$ and $\tilde{A}^T$ are the same, we may apply the same proof steps to $\tilde{A}^T$ instead of $\tilde{A}$. In this case we obtain the analogous result $\kappa(A) \geq \kappa_B(K)\max\{1, \frac{\|B\|_2}{\sigma_{1,B}}, \frac{\|\tilde{B}\|_2}{\sigma_{1,B}}\}$ and we conclude by taking the maximum of the two lower bounds. □

References

[1] T. McDevitt, T. Laursen, A mortar-finite element formulation for frictional contact problems, Int. J. Numer. Methods Biomed. Eng. 48 (10) (2000) 1525–1547.
[2] M.A. Puso, T.A. Laursen, A mortar segment-to-segment contact method for large deformation solid mechanics, Comput. Methods Appl. Mech. Eng. 193 (6–8) (2004) 601–629.
[3] A. Popp, Mortar methods for computational contact mechanics and general interface problems, Ph.D. thesis, Technische Universität München, 2012.
[4] S. Deparis, D. Forti, P. Gervasio, A. Quarteroni, INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces, Comput. Fluids 141 (2016) 22–41.
[5] P. Gervasio, A. Quarteroni, Analysis of the INTERNODES method for non-conforming discretizations of elliptic equations, Comput. Methods Appl. Mech. Eng. 334 (2018) 138–166, https://doi.org/10.1016/j.cma.2018.02.004.
[6] S. Deparis, P. Gervasio, Conservation of forces and total work at the interface using the INTERNODES method, Vietnam J. Math. (2022), https://doi.org/10.1007/s10013-022-00560-9, in press.
[7] O. Günther-Hanssen, Finite element method INTERNODES for contact mechanics. A study on condition number and iterative solver performance, Infoscience EPFL (2020), http://infoscience.epfl.ch/record/278135.
[8] S. Deparis, D. Forti, A. Quarteroni, A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids, SIAM J. Sci. Comput. 36 (6) (2014) A2745–A2762.
[9] J.A. Freeman, D. Saad, Learning and generalization in radial basis function networks, Neural Comput. 7 (5) (1995) 1000–1020.
[10] H. Wendland, Meshless Galerkin methods using radial basis functions, Math. Comput. 68 (228) (1999) 1521–1531.
[11] J. Wang, G. Liu, A point interpolation Meshless method based on radial basis functions, Int. J. Numer. Methods Biomed. Eng. 54 (11) (2002) 1623–1648.
[12] M.D. Buhmann, Radial basis functions, Acta Numer. 9 (2000) 1–38.
[13] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1) (1995) 389–396.
[14] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, 2012.
[15] Y. Voet, On the preconditioning of the INTERNODES matrix for applications in contact mechanics, Master's thesis, École polytechnique féférale de Lausanne, 2021, http://infoscience.epfl.ch/record/289288.
[16] N. Kikuchi, Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods, SIAM Studies in Applied Mathematics, vol. 8, SIAM, Philadelphia, Pa, 1988.
[17] M. Sofonea, Mathematical Models in Contact Mechanics, London Mathematical Society Lecture Note Series, vol. 398, Cambridge University Press, New York, 2012.
[18] P. Wriggers, Computational Contact Mechanics, 2nd edition, Springer, Berlin, Heidelberg, 2006.
[19] A. Curnier, Unilateral contact, in: New Developments in Contact Problems, Springer, 1999, pp. 1–54.

[20] W. Karush, Minima of functions of several variables with inequalities as side conditions, Master's thesis, Department of Mathematics, University of Chicago, Chicago, IL, USA, 1939.
[21] H. Kuhn, A. Tucker, Nonlinear programming, in: Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, 1951, pp. 481–492.
[22] L.E. Malvern, Introduction to the Mechanics of a Continuous Medium, Prentice-Hall Series in Engineering of the Physical Sciences, Prentice-Hall, Englewood Cliffs N.J, 1969.
[23] F. Brezzi, M. Fortin, Mixed and Hybrid Finite Element Methods, Springer Series in Computational Mathematics, vol. 15, Springer, New York, NY, 1991.
[24] P. Gervasio, A. Quarteroni, The INTERNODES method for non-conforming discretizations of PDEs, Commun. Appl. Math. 1 (2019) 361–401, https://doi.org/10.1007/s42967-019-00020-1.
[25] P. Bochev, R. Lehoucq, Energy principles and finite element methods for pure traction linear elasticity, Comput. Methods Appl. Math. 11 (2) (2011) 173–191.
[26] M. Benzi, M.A. Olshanskii, An augmented Lagrangian-based approach to the Oseen problem, SIAM J. Sci. Comput. 28 (6) (2006) 2095–2113.
[27] M. Benzi, M.A. Olshanskii, Z. Wang, Modified augmented Lagrangian preconditioners for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 66 (4) (2011) 486–508.
[28] S. Deparis, G. Grandperrin, A. Quarteroni, Parallel preconditioners for the unsteady Navier–Stokes equations and applications to hemodynamics simulations, Comput. Fluids 92 (2014) 253–273.
[29] J.W. Pearson, A.J. Wathen, A new approximation of the Schur complement in preconditioners for PDE-constrained optimization, Numer. Linear Algebra Appl. 19 (5) (2012) 816–829.
[30] C. Greif, D. Schötzau, Preconditioners for saddle point linear systems with highly singular (1, 1) blocks, in: Special Volume on Saddle Point Problems, Electron. Trans. Numer. Anal. 22 (2006) 114–121.
[31] C. Greif, D. Schötzau, Preconditioners for the discretized time-harmonic Maxwell equations in mixed form, Numer. Linear Algebra Appl. 14 (4) (2007) 281–297.
[32] I. Perugia, V. Simoncini, Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations, Numer. Linear Algebra Appl. 7 (7–8) (2000) 585–616.
[33] M.F. Adams, Algebraic multigrid methods for constrained linear systems with applications to contact problems in solid mechanics, Numer. Linear Algebra Appl. 11 (2–3) (2004) 141–153.
[34] A. Franceschini, M. Ferronato, M. Frigo, C. Janna, A reverse augmented constraint preconditioner for Lagrange multiplier methods in contact mechanics, Comput. Methods Appl. Mech. Eng. 392 (2022) 114632.
[35] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[36] Z.-H. Cao, Augmentation block preconditioners for saddle point-type matrices with singular (1, 1) blocks, Numer. Linear Algebra Appl. 15 (6) (2008) 515–533.
[37] G.H. Golub, C. Greif, On solving block-structured indefinite linear systems, SIAM J. Sci. Comput. 24 (6) (2003) 2076–2092.
[38] Q. Liu, New preconditioners for nonsymmetric saddle point systems with singular (1, 1) block, Int. Sch. Res. Not. (2013) 2013.
[39] D. Kressner, Computational Linear Algebra, Lecture Notes, 2020.
[40] Y. Saad, Numerical Methods for Large Eigenvalue Problems, revised edition, SIAM, 2011.
[41] M. Benzi, J. Liu, Block preconditioning for saddle point systems with indefinite (1, 1) block, Int. J. Comput. Math. 84 (8) (2007) 1117–1129.
[42] Y. Saad, Iterative Methods for Sparse Linear Systems, SIAM, 2003.
[43] A. Ern, J.-L. Guermond, Theory and Practice of Finite Elements, vol. 159, Springer, 2004.
[44] A.J. Wathen, Preconditioning, Acta Numer. 24 (2015) 329–376.
[45] T.A. Davis, Direct Methods for Sparse Linear Systems, SIAM, 2006.
[46] M. Benzi, M. Tůma, A robust incomplete factorization preconditioner for positive definite matrices, Numer. Linear Algebra Appl. 10 (5–6) (2003) 385–400.